



**TUGAS AKHIR-TE 141599**

**IMPLEMENTASI JARINGAN SENSOR NIRKABEL UNTUK  
PEMANTAUAN ELECTROCARDIOGRAM DI TENDA  
MEDIS MENGGUNAKAN RASPBERRY PI**

Dwi Sandi Wijayanto  
NRP 2213100077

Dosen Pembimbing  
Dr. Achmad Arifin, S.T., M.Eng.  
Dr. Tri Arief Sardjono, S.T., M.T.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





**FINAL PROJECT-TE 141599**

**IMPLEMENTATION OF WIRELESS SENSOR NETWORK  
TO MONITORING ELECTROCARDIOGRAM IN MEDICAL  
TENT USING RASPBERRY PI**

Dwi Sandi Wijayanto  
NRP 2213100077

Advisor  
Dr. Achmad Arifin, S.T., M.Eng.  
Dr. Tri Arief Sardjono, S.T., M.T.

ELECTRICAL ENGINEERING DEPARTMENT  
Faculty of Electrical Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017

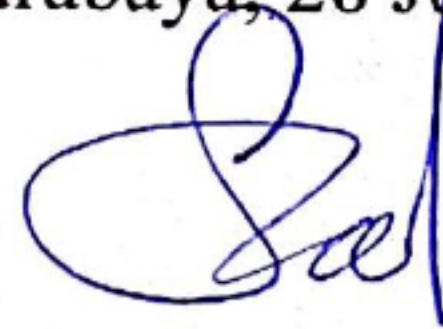


## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “Implementasi Jaringan Sensor Nirkabel untuk Pemantauan Elektrokardiogram di Tenda Medis Menggunakan Raspberry Pi” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 28 Juli 2017



Dwi Sandi Wijayanto

2213100077





**IMPLEMENTASI JARINGAN SENSOR NIRKABEL UNTUK  
PEMANTAUAN ELEKTROKARDIOGRAM DI TENDA MEDIS  
MENGUNAKAN RASPBERRY PI**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada**

**Bidang Studi Elektronika  
Departemen Teknik Elektro  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember**

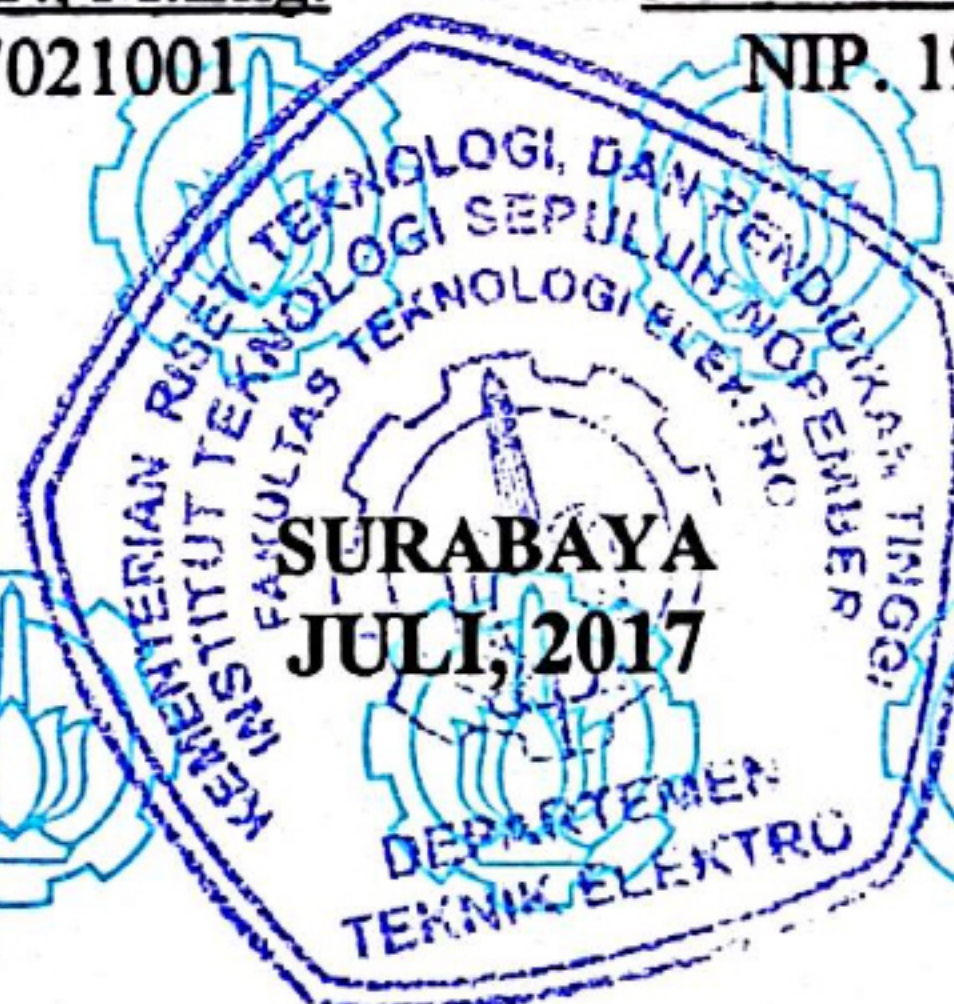
**Menyetujui:**

**Dosen Pembimbing I**

**Dosen Pembimbing II**

**Dr. Achmad Arifin, S.T., M.Eng.**  
**NIP. 197103141997021001**

**Dr. Tri Arief Sardjono, S.T., M.T.**  
**NIP. 197002121995121001**









# **IMPLEMENTASI JARINGAN SENSOR NIRKABEL UNTUK PEMANTAUAN ELEKTROKARDIOGRAM DI TENDA MEDIS MENGGUNAKAN RASPBERRY PI**

**Nama : Dwi Sandi Wijayanto**  
**Pembimbing I : Dr. Achmad Arifin, S.T., M.Eng.**  
**Pembimbing II : Dr. Tri Arief Sardjono, S.T., M.T.**

## **ABSTRAK**

Banyaknya korban bencana alam di tenda medis tidak diimbangi dengan sumber daya seperti ahli dan alat medis. Hal ini menimbulkan masalah yang signifikan. Dengan jumlah tenaga medis yang sedikit, maka dibutuhkan waktu dan tenaga yang lebih banyak untuk memantau beberapa korban bencana alam di tenda medis, khususnya di tenda ICU. Oleh karena itu dibutuhkan suatu sistem yang dapat memantau kondisi para pasien korban bencana alam tersebut. Sistem ini bekerja dengan menangkap sinyal jantung melalui elektroda, kemudian diamplifikasi sebesar 1000 kali dan diband pass filter dari 0.05 Hz sampai 40 Hz, untuk didapatkan sinyal listrik jantung. Sinyal listrik jantung masuk ke ADC Arduino uno untuk dikonversi ke sinyal digital. Data ADC kemudian dikirimkan secara wireless ke server atau receiver menggunakan NRF24L01. Menggunakan beberapa NRF24L01 sebagai *transmitter* atau *client* dan 1 NRF24L01 sebagai *receiver* atau *server base station*. Data yang diterima *base station* dikirim serial ke Raspberry Pi untuk diolah dan dibuat GUI atau *interface*. *Interface* di desktop Raspberry Pi ini memudahkan untuk memonitoring elektrokardiogram dan *heart rate* banyak pasien secara langsung. Setelah dilakukan pengujian jarak cakupan NRF24L01 diruangan terbuka dan tanpa halangan, didapatkan loss data pada jarak 2 m sampai 18 m sebesar 0%, pada jarak 24 m sebesar 1.7%, pada jarak 30 m sebesar 0 %, dan pada jarak 36 m sebesar 1.5%. Sinyal kemudian di olah secara digital menggunakan algoritma pan tompkin dan interval RR untuk dihitung heart rate masing masing ekg. Untuk penelitian kedepannya dapat digunakan ekg, mikrokontroler, beserta LCD display yang lebih kecil.

Kata kunci: Raspberry Pi, *electrocardiogram*, jantung, ADC Arduino Uno, NRF24L01, Jaringan Sensor Nirkabel.

*Halaman ini sengaja dikosongkan*

# **IMPLEMENTASI JARINGAN SENSOR NIRKABEL UNTUK PEMANTAUAN ELECTROCARDIOGRAM DI TENDA MEDIS USING RASPBERRY PI**

**Name** : Dwi Sandi Wijayanto  
**Advisor I** : Dr. Achmad Arifin, S.T., M.Eng.  
**Advisor II** : Dr. Tri Arief Sardjono, S.T., M.T.

## **ABSTRACT**

*The number of victims of natural disasters in medical tents is not balanced with resources such as experts and medical devices. This poses a significant problem. With a small number of medical personnel, it will take more time and effort to monitor some victims of natural disasters in medical tents, especially in ICU tents. Therefore, a system that can monitor the condition of the victims of the natural disaster is needed. This system works by capturing heart signals through electrodes, then amplified 1000 times and filtered from 0.05 Hz to 40 Hz, to obtain electrical heart signals. The heart's electrical signals enter the Arduino uno ADC to convert to digital signals. The ADC data is then sent to the server or receiver using NRF24L01. Use multiple NRF24L01 as transmitter or client and 1 NRF24L01 as receiver or base station server. Data received base station sent serial to Raspberry Pi for processing and made GUI or interface. The Raspberry Pi desktop interface makes it easy to monitor multiple patient electrocardiograms and heart rate directly. After testing the NRF24L01 coverage in open area and without obstacle, data loss was obtained from distance 2 m to 18 m was 0 %, 24 m distance was 1.7%, 30 m distance was 0%, and at 36 m distance was 1.5%. The signal then digitally calculated by the pan tompkin algorithm and RR interval to count the heart rate of each ekg. For future research can use a smaller ekg, microcontroller, and LCD display.*

**Keywords:** Raspberry Pi, electrocardiogram, heart, ADC ArduinoUno, NRF24L01, Wireless Sensor Network.



*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan tugas akhir yang berjudul **Implementasi Jaringan Sensor Nirkabel untuk Pemantauan Elektrokardiogram di Tenda Medis Menggunakan Raspberry Pi**. Tugas akhir ini dibuat sebagai persyaratan akademis program studi sarjana di Jurusan Teknik Elektro ITS. Tugas akhir ini juga adalah kesempatan untuk mengaplikasikan teori dan ilmu yang telah diperoleh selama menempuh pendidikan sarjana untuk menghasilkan karya yang bermanfaat untuk kemajuan teknologi.

Oleh karena itu, penulis sampaikan terima kasih kepada semua pihak yang telah membantu sehingga tugas akhir ini dapat selesai tepat waktu tanpa halangan yang berarti. Dengan hormat dan rendah diri penulis sampaikan terima kasih sebesar besarnya kepada:

1. Keluarga penulis, Sukatmidjo, Kartini, dan Evi Mijayanti yang telah mendukung penulis secara moral dan materiil .
2. Dosen Pembimbing, Bapak Dr. Achmad Arifin, S.T., M.Eng. dan Bapak Dr. Tri Arief, S.T., M.Eng yang telah mendidik, membantu dan mengarahkan penulis hingga terselesaikannya tugas akhir ini.
3. Alumni dan senior yang membantu dalam pengerjaan alat.
4. Teman teman mahasiswa teknik elektro ITS, sebagai rekan rekan yang membantu hingga tugas akhir ini dapat selesai tepat pada waktunya dengan hasil yang memuaskan.

Besar harapan penulis agar tugas akhir ini dapat menjadi acuan untuk pengembangan teknologi elektronika medis, khususnya di bidang jantung. Penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna. Oleh karena itu penulis memohon maaf apabila terdapat kesalahan, dan penulis menerima masukan, kritik, serta saran dari pembaca sekalian. Akhir kata, semoga tugas akhir ini dapat berguna bagi pembaca. Terima kasih.

Surabaya, 28 Juli 2017

Penulis

*Halaman ini sengaja dikosongkan*



## DAFTAR ISI

<b>ABSTRAK</b> .....	i
<b>ABSTRACT</b> .....	iii
<b>KATA PENGANTAR</b> .....	v
<b>DAFTAR ISI</b> .....	vii
<b>TABLE OF CONTENT</b> .....	xi
<b>DAFTAR GAMBAR</b> .....	xv
<b>DAFTAR TABEL</b> .....	xvii
<b>BAB I</b> .....	1
<b>PENDAHULUAN</b> .....	1
<b>1.1. Latar Belakang</b> .....	1
<b>1.2. Perumusan Masalah</b> .....	2
<b>1.3. Tujuan</b> .....	2
<b>1.4. Metodologi</b> .....	2
<b>1.5. Sistematika Penulisan</b> .....	3
<b>1.6. Relevansi</b> .....	4
<b>BAB II</b> .....	5
<b>TEORI PENUNJANG</b> .....	5
<b>2.1. Kelistrikan Jantung</b> .....	5
<b>2.1.1. Biopotensial</b> .....	5
<b>2.1.2. Sinyal Jantung</b> .....	6
<b>2.1.3. Elektrokardiogram</b> .....	10
<b>2.2. Elektrokardiograf</b> .....	10
<b>2.2.1. Sadapan EKG</b> .....	11
<b>2.2.2. Penguat Instrumentasi</b> .....	12
<b>2.2.3. High Pass Filter</b> .....	15
<b>2.2.4. Low Pass Filter</b> .....	17
<b>2.2.5. Band Stop Filter</b> .....	19
<b>2.2.6. Rangkaian Penjumlah</b> .....	22
<b>2.3. Pemrosesan Sinyal Digital</b> .....	23

2.3.1.	Arduino Uno .....	24
2.3.2.	NRF24L01 .....	25
2.3.3.	Komunikasi SPI .....	26
2.3.4.	Raspberry Pi.....	26
2.3.5.	Pengolahan Sinyal Digital .....	27
<b>BAB III</b>	.....	31
<b>PERANCANGAN SISTEM</b>	.....	31
2.4.	Perancangan Perangkat Elektrokardiogram.....	31
3.1.1.	Perancangan Penguat Instrumentasi .....	32
3.1.2.	Perancangan <i>High Pass Filter</i> .....	36
3.1.3.	Perancangan <i>Low Pass Filter</i> .....	37
3.1.4.	Perancangan <i>Band Stop Filter</i> .....	39
3.1.5.	Perancangan Rangkaian Penjumlah .....	40
3.1.6.	Perancangan Rangkaian Proteksi .....	40
3.2.	Perancangan Sistem Piranti Lunak .....	42
3.2.1.	Perancangan <i>Analog To Digital Converter</i> .....	42
3.2.2.	Perancangan Transmisi Data Multinode NRF24L01 .....	42
3.2.2.1	Perancangan Program Transmitter 1 .....	43
3.2.2.2	Perancangan Program Transmitter 2 .....	43
3.2.2.3	Program Receiver .....	45
3.2.3.	Raspberry Pi.....	46
3.2.4.	Perancangan Deteksi Segmentasi .....	46
3.2.5.	Perancangan Komputasi Data .....	49
<b>BAB IV</b>	.....	55
<b>PENGUJIAN DAN ANALISIS</b>	.....	55
4.1.	Pengujian Perangkat Elektrokardiogram.....	55
4.1.1.	Pengujian Instrumentation Amplifier IC AD620.....	55
4.1.2.	Pengujian <i>HighPass Filter</i> .....	56
4.1.3.	Pengujian <i>Low Pass Filter</i> .....	57

4.1.4.	Pengujian <i>Band Stop Filter</i> .....	60
4.1.5.	Pengujian Rangkaian Penjumlah .....	61
4.2.	Pengujian Sistem Piranti Lunak .....	64
4.2.1.	Pengujian Jarak Cakupan NRF24L01 .....	64
4.2.2.	Pengujian Multinode (2 Transmitter dan 1 Receiver) .....	64
4.2.3.	Pengujian Komputasi Data .....	65
4.2.4.	Pengujian Sistem Keseluruhan .....	68
BAB V	.....	71
PENUTUP	.....	71
5.1.	Kesimpulan .....	71
5.2.	Saran.....	71
DAFTAR PUSTAKA	.....	753
LAMPIRAN	.....	75
BIODATA PENULIS	.....	87



*Halaman ini sengaja dikosongkan*

## **TABLE OF CONTENT**

<b>ABSTRAK</b> .....	i
<b>ABSTRACT</b> .....	iii
<b>PREFACE</b> .....	v
<b>TABLE OF CONTENT</b> .....	vii
<b>TABLE OF CONTENT</b> .....	xi
<b>LIST OF FIGURES</b> .....	xv
<b>LIST OF TABLES</b> .....	xvii
<b>CHAPTER I</b> .....	1
<b>INTRODUCTION</b> .....	1
<b>1.1. Background of Study</b> .....	1
<b>1.2. Statement of Problems</b> .....	2
<b>1.3. Goals</b> .....	2
<b>1.4. Methodology</b> .....	2
<b>1.5. Structure of the Study</b> .....	3
<b>1.6. Relevances</b> .....	4
<b>CHAPTER II</b> .....	5
<b>SUPPORTING THEORY</b> .....	5
<b>2.1. Electrical Heart</b> .....	5
<b>2.1.1. Biopotential</b> .....	5
<b>2.1.2. Heart Signal</b> .....	6
<b>2.1.3. Electrocardiogram</b> .....	10
<b>2.2. Electrocardiograph</b> .....	10
<b>2.2.1. ECG Leads</b> .....	11
<b>2.2.2. Instrumentation Amplifier</b> .....	12
<b>2.2.3. High Pass Filter</b> .....	15
<b>2.2.4. Low Pass Filter</b> .....	17
<b>2.2.5. Band Stop Filter</b> .....	19
<b>2.2.6. Adder</b> .....	22
<b>2.3. Analog Signal Processing</b> .....	23

2.3.1.	<i>Arduino Uno</i> .....	24
2.3.2.	<i>NRF24L01</i> .....	25
2.3.3.	<i>SPI Communication</i> .....	26
2.3.4.	<i>Raspberry Pi</i> .....	26
2.3.5.	<i>Digital Signal Processing</i> .....	27
<b>CHAPTER III</b> .....		31
<b>DESIGN OF THE SYSTEM</b> .....		31
2.4.	<i>Design of the Electrocardiograph</i> .....	31
3.1.1.	<i>Design of Instrumentation Amplifier</i> .....	32
3.1.2.	<i>Design of High Pass Filter</i> .....	36
3.1.3.	<i>Design of Low Pass Filter</i> .....	37
3.1.4.	<i>Design of Band Stop Filter</i> .....	39
3.1.5.	<i>Design of Adder</i> .....	40
3.1.6.	<i>Design of Protection Circuit</i> .....	40
3.2.	<i>Design of the Software</i> .....	42
3.2.1.	<i>Design of Analog To Digital Converter</i> .....	42
3.2.2.	<i>Design of Multinode Data Transmission NRF24L01</i> .....	42
3.2.2.1	<i>Design of Transmitter 1 Program</i> .....	43
3.2.2.2	<i>Design of Transmitter 2 Program</i> .....	43
3.2.2.3	<i>Receiver Program</i> .....	45
3.2.3.	<i>Raspberry Pi</i> .....	46
3.2.4.	<i>Design of Segmentation Detection</i> .....	46
3.2.5.	<i>Design of Data Computation</i> .....	49
<b>CHAPTER IV</b> .....		55
<b>PENGUJIAN DAN ANALISIS</b> .....		55
4.1.	<i>Evaluation of the Electrocardiograph</i> .....	55
4.1.1.	<i>Evaluation of Instrumentation Amplifier AD620</i> .....	55
4.1.2.	<i>Evaluation of HighPass Filter</i> .....	56
4.1.3.	<i>Evaluation of Low Pass Filter</i> .....	57

4.1.4.	<i>Evaluation of Band Stop Filter .....</i>	60
4.1.5.	<i>Evaluation of Adder Circuit .....</i>	61
4.2.	<i>Evaluation of the Software Design .....</i>	64
4.2.1.	<i>Evaluation of Distance Coverage NRF24L01 .....</i>	64
4.2.2.	<i>Evaluation of Multinode (2 Transmitter and 1 Receiver)</i> <i>64</i>	
4.2.3.	<i>Evaluation of Computation Data .....</i>	65
4.2.4.	<i>Evaluation of the Whole System .....</i>	68
CHAPTER V	.....	71
CONCLUSION	.....	71
5.1.	<i>Conclusion .....</i>	71
5.2.	<i>Suggestion .....</i>	71
BIBLIOGRAPHY	.....	753
APPENDIX	.....	75
CURRICULUM VITAE	.....	87

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

Gambar 1. 1 Statistik Bencana di Indonesia 2017.....	1
Gambar 2. 1 Terjadinya Potensial Aksi .....	6
Gambar 2. 2 Anatomi Nodus Jantung .....	8
Gambar 2. 3 Morfologi Sinyal Jantung .....	8
Gambar 2. 4 Pembentukan Satu Siklus Sinyal Jantung.....	9
Gambar 2. 5 Sinyal Elektrokardiogram Beserta Komponennya .....	10
Gambar 2. 6 Prinsip Kerja Elektrokardiograf .....	11
Gambar 2. 7 EKG konvensional .....	12
Gambar 2. 8 Segitiga Einthoven dan Sadapan Standar .....	13
Gambar 2. 9 Rangkaian Penguat Instrumentasi .....	14
Gambar 2. 10 Rangkaian HPF -40dB/dec .....	16
Gambar 2. 11 Rangkaian LPF -40dB/dec .....	18
Gambar 2. 12 Rangkaian BPF -40dB.....	20
Gambar 2. 13 Rangkaian Penjumlah .....	23
Gambar 2. 14 Arduino Uno R3 .....	24
Gambar 2. 15 NRF24L01 .....	26
Gambar 2. 16 Raspberry Pi .....	27
 Gambar 3. 1 Diagram Blok Rangkaian EKG.....	 31
Gambar 3. 2 Diagram Blok Sistem Keseluruhan .....	32
Gambar 3. 3 Penempatan Elektroda.....	33
Gambar 3. 4 Filter <i>Low Pass</i> Pasif.....	33
Gambar 3. 5 Rancangan Rangkaian Instrumentasi .....	35
Gambar 3. 6 Rancangan Rangkaian HPF.....	36
Gambar 3. 7 Respon Frekuensi HPF.....	36
Gambar 3. 8 Rancangan Rangkaian LPF .....	38
Gambar 3. 9 Respon Frekuensi LPF .....	38
Gambar 3. 10 Rancangan Rangkaian BPF .....	39
Gambar 3. 11 Respon Frekuensi Rangkaian BPF .....	39
Gambar 3. 12 Rancangan Rangkaian Penjumlah .....	41
Gambar 3. 13 Rangkaian Proteksi RC dan Dioda .....	41
Gambar 3. 14 Diagram Blok Sistem Piranti Lunak.....	43
Gambar 3. 15 Flowchart program transmitter 1 .....	44
Gambar 3. 16 Flowchart program transmitter 2 .....	45
Gambar 3. 17 Flowchart program receiver .....	47
Gambar 3. 18 Flowchart program GUI <i>QT Creator</i> .....	48



Gambar 3. 19 Flowchart perhitungan heart rate .....	53
Gambar 4. 1 Hasil Pengujian <i>Instrumentation Amplifier</i> .....	56
Gambar 4. 2 Rangkaian Pengujian <i>High Pass Filter</i> .....	57
Gambar 4. 3 Grafik Pengujian <i>High Pass Filter</i> .....	58
Gambar 4. 4 Pengukuran HPF saat frekuensi terendah di function generator (0.0798 Hz) .....	58
Gambar 4. 5 Rangkaian Pengujian <i>Low Pass Filter</i> .....	59
Gambar 4. 6 Grafik Pengujian Rangkaian Low Pass Filter .....	59
Gambar 4. 7 Rangkaian Pengujian <i>Band Stop Filter</i> .....	60
Gambar 4. 8 Grafik Pengujian Rangkaian Band Stop Filter .....	62
Gambar 4. 9 Pengujian BSF ketika frekuensi 50Hz .....	62
Gambar 4. 10 Rangkaian Pengujian Adder .....	63
Gambar 4. 11 Hasil Pengujian Rangkaian Penjumlah .....	63
Gambar 4. 12 Flowchart pengujian jarak cakupan NRF24L01 .....	65
Gambar 4. 13 Serial monitor receiver multinode pada jarak 12 meter .....	67
Gambar 4. 14 Pengujian Algoritma Pan Tomkin di Delphi .....	68
Gambar 4. 15 GUI Multinode EKG Monitoring di Raspberry Pi .....	69

## DAFTAR TABEL

Tabel 4. 1 Pengujian <i>High Pass Filter</i> .....	57
Tabel 4. 2 Pengujian <i>Low Pass Filter</i> .....	60
Tabel 4. 3 Pengujian Rangkaian <i>Band Stop Filter</i> .....	61
Tabel 4. 4 Pengujian jarak cakupan NRF24L01.....	66

*Halaman ini sengaja dikosongkan*

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

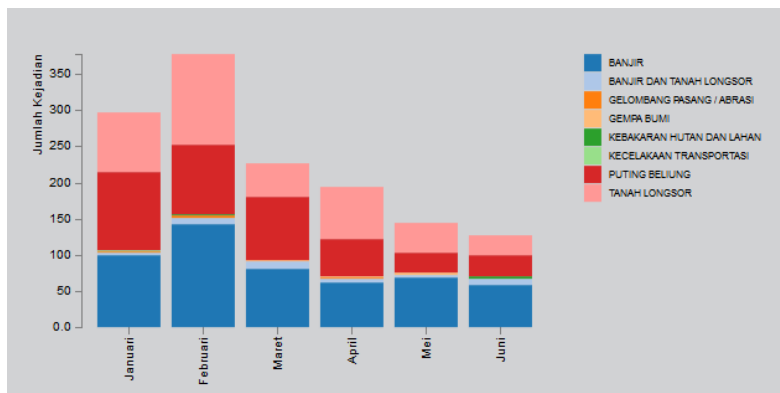
Bencana alam yang menimpa suatu daerah, seringkali menimbulkan korban jiwa dan kerusakan, baik itu korban meninggal, korban luka luka maupun kerusakan fasilitas umum dan harta benda masyarakat. Menurut Badan Nasional Penanggulangan Bencana (BNPB), hingga bulan Juni 2017, data bencana yang tercatat adalah:

1. Jumlah kejadian : 1.368
2. Korban meninggal dan hilang : 227
3. Korban menderita dan mengungsi : 1.710.973
4. Kerusakan permukiman : 18.983

Statistik bencana di Indonesia dapat dilihat di Gambar 1.1[1].

Terjadinya bencana alam sering mengakibatkan wilayah terkena dampak menjadi terisolasi sehingga sulit dijangkau oleh para relawan untuk memberikan pertolongan dan bantuan[1].

Penanganan medis lebih difokuskan pada penanganan medis saat pasien telah tiba di rumah sakit karena tenaga medis ditempatkan secara sentral sehingga pasien harus mendatangi langsung lokasi untuk memperoleh pemeriksaan dan penanganan medis. Khusus di daerah bencana, banyaknya korban bencana alam di tenda medis tidak diimbangi dengan sumber daya seperti ahli dan alat medis. Hal ini menimbulkan masalah yang signifikan. Dengan jumlah tenaga medis yang sedikit, maka



Gambar 1.1. Statistik Bencana di Indonesia 2017

dibutuhkan waktu yang lebih lama untuk memantau seluruh korban bencana alam. Oleh karena itu dibutuhkan suatu sistem yang dapat memantau kondisi para pasien korban bencana alam tersebut.

Oleh karena itu diperlukan adanya sistem monitoring electrocardiogram berbasis *wireless sensor network* yang bisa memonitoring elektrokardiogram pasien secara langsung, dimana tenaga medis bisa memantau kondisi electrokardiogram dari beberapa pasien secara satu waktu dan disatu tampilan layar.

## **1.2. Perumusan Masalah**

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa permasalahan utama dalam tugas akhir ini, yaitu:

1. Kurangnya tenaga dan alat medis di daerah bencana alam.
2. Membutuhkan banyak tenaga dan alat medis untuk memantau kondisi para pasien di tenda medis
3. Tidak sedikit pasien yang tidak terpantau kondisinya karena tidak adanya atau kurangnya alat monitoring

Adapun batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Merancang elektrokardiogram dengan peletakan elektroda memanfaatkan metode Einthoven dengan tiga lead yang ditempatkan di dada dengan elektroda tempel.
2. Mengakuisisi data dari beberapa elektrokardiogram secara *Wireless Sensor Network*
3. Pasien yang di pantau berada dalam satu tenda medis.

## **1.3. Tujuan**

Penelitian tugas akhir ini diharapkan mencapai tujuan berikut:

1. Menemukan metode yang tepat untuk membuat akuisisi data dari beberapa elektrokardiogram pasien secara *real time*.
2. Memantau elektrokardiogram beberapa pasien di tenda medis.

## **1.4. Metodologi**

Dalam penyelesaian tugas akhir ini digunakan metodologi sebagai berikut:

1. Studi Literatur

Pada tahap studi literatur dilakukan pengumpulan referensi yang menunjang pembuatan alat dan penulisan tugas akhir. Referensi

diambil dari buku teks kedokteran dan elektronika, serta dari jurnal, dan artikel konferensi.

2. Perancangan Perangkat Keras

Pada tahapan ini dirancang dan direalisasikan perangkat keras yang terdiri dari rangkaian elektrokardiogram, ADC Arduino Uno, Raspberry Pi.

3. Perancangan Perangkat Lunak

Pada tahap ini perangkat lunak didesain untuk mengolah sinyal jantung, membuat pengolahan akuisisi data multinode EKG dan pembuatan GUI di Raspberry Pi.

4. Pengujian Sistem

Tahap pengujian sistem dilakukan untuk menguji kehandalan dan keakuratan sistem, mengambil data sampel pengguna, dan membandingkan dengan sistem yang sudah ada.

5. Penulisan Laporan Tugas Akhir

Tahap terakhir adalah penulisan laporan yang bertujuan sebagai bentuk pertanggung jawaban dan dokumentasi. Laporan tugas akhir berisi pendahuluan, teori-teori penunjang yang digunakan, perancangan sistem, pengujian sistem, dan penutup yang berisi kesimpulan dan saran.

## 1.5. Sistematika Penulisan

Laporan ini disusun dengan sistematika sebagai berikut:

### **BAB I PENDAHULUAN**

Pada bab ini dibahas latar belakang, perumusan masalah, batasan masalah, tujuan, metodologi, sistematika pembahasan, dan relevansi dari tugas akhir.

### **BAB II TEORI PENUNJANG**

Bab ini berisi dasar teori dan literatur yang digunakan sebagai penunjang tugas akhir.

### **BAB III PERANCANGAN SISTEM**

Pada bab ini dibahas rancangan sistem secara mendetail, mulai dari desain perangkat keras, perangkat lunak, dari dasar algoritma hingga aplikasinya

### **BAB IV PENGUJIAN DAN ANALISA**

Pada bab ini dipaparkan hasil pengambilan data dan pengujian, yang disertai dengan analisa yang berkaitan.



## **BAB V        PENUTUP**

Bab ini berisi kesimpulan yang diperoleh dari pembuatan tugas akhir ini, serta saran-saran untuk pengembangan lebih lanjut.

### **1.6. Relevansi**

Hasil yang dari tugas akhir ini diharapkan dapat menjadi referensi untuk pembuatan monitoring elektrokardiogram dengan jaringan sensor nirkabel yang real time.

## **BAB II**

### **TEORI PENUNJANG**

Pada Bab II dibahas tinjauan pustaka dan dasar teori penunjang yang digunakan sebagai dasar pembuatan tugas akhir ini. Bab ini juga berisi sistem dan algoritma perangkaan tugas akhir ini dan tugas akhir yang pernah dibuat sebelumnya. Teori penunjang dibagi menjadi 3, anatomi dan kelistrikan jantung; pemrosesan sinyal analog, dan pemrosesan sinyal digital.

#### **2.1. Kelistrikan Jantung**

Jantung adalah salah satu organ tubuh vital pada manusia, yang bertugas untuk memompa darah ke seluruh tubuh. Organ ini memiliki sinyal yang dapat menunjukkan informasi jantung tersebut. Sinyal informasi ini dalam bentuk sinyal elektrik.

##### **2.1.1. Biopotensial**

Biopotensial adalah tegangan yang dihasilkan oleh jaringan tubuh, terutama jaringan otot selama berkontraksi, atau potensial listrik yang terjadi antara titik dalam sel, jaringan, atau organisme hidup. Potensial aksi adalah sebuah mekanisme dari neuron-neuron yang terdapat di dalam otak untuk mengantar informasi otak dan memprosesnya kembali. Hubungan antar neuron ini tidak hanya sebatas aliran listrik yang terjadi secara otomatis, namun, listrik yang dihasilkan berasal dari hasil difusi senyawa  $\text{Na}^+$  dan  $\text{K}^+$  yang berada didalam Akson, yang dapat dilihat pada Gambar 2.1.

Di dalam membran Akson terdapat konsentrasi  $\text{K}^+$  yang lebih tinggi daripada konsentrasi  $\text{Na}^+$  dan di luar membran akson terdapat kadar  $\text{Na}^+$  yang lebih tinggi. Perbedaan konsentrasi dari kedua substrat tersebut membuat adanya difusi dari dalam membran keluar dan sebaliknya. Setiap terjadinya perubahan konsentrasi, maka akan terjadi sebuah lonjakan listrik pada akson tersebut. Pada dinding membran tersebut juga terdapat gerbang-gerbang ion  $\text{Na}^+$  dan  $\text{K}^+$ . Dalam kondisi normal, dimana konsentrasi  $\text{K}^+$  lebih tinggi di dalam membran, maka terdapat tegangan sebesar -70mV, yang kemudian disebut sebagai kondisi polarisasi.

Ketika terjadi sebuah impuls, gerbang  $\text{Na}^+$  akan terbuka dan menyebabkan konsentrasi  $\text{Na}^+$  di dalam akson bertambah seiring dengan ion-ion  $\text{Na}^+$  yang masuk kedalam akson. Kondisi ini akan membuat

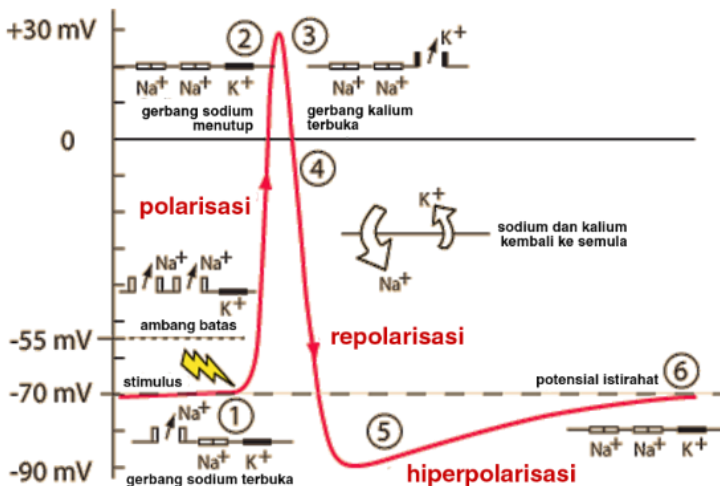
membran akson menjadi lebih positif dibandingkan diluar membrannya. Fenomena ini disebut depolarisasi, dimana potensial antar membrane tersebut akan naik dari  $-70\text{mV}$  mencapai titik maksimum di  $+40\text{mV}$ .

Ketika konsentrasi  $\text{Na}^+$  sudah tinggi, maka gerbang-gerbang  $\text{K}^+$  akan membuka sehingga ion  $\text{K}^+$  akan keluar dari membran. Hal ini yang menyebabkan kondisi didalam membran kembali menjadi lebih negatif daripada diluar membran dan menyebabkan repolarisasi.

Namun gerbang  $\text{K}^+$  ini membuka dan menutup dengan durasi lebih lama, sehingga terdapat ion  $\text{K}^+$  yang terbuang terlebih, menyebabkan nilai tegangan antar neuron tersebut lebih negatif dari sebelumnya, yang disebut sebagai hiperpolarisasi. Untuk mengembalikan ke keadaan semula, kedua ion yaitu ion  $\text{K}^+$  dan ion  $\text{Na}^+$  akan berdifusi keluar dan masuk sehingga akan menghasilkan nilai yang sama dengan awalnya. Aktivitas ini terjadi di nodus Ranvier, yaitu lokasi dimana akson tidak diselubungi oleh selaput myelin [12].

### 2.1.2. Sinyal Jantung

Jantung bergerak berdasarkan sinyal yang dikirimkan secara tak sadar. Sinyal ini berupa impuls kelistrikan jantung. Impuls tersebut berasal dari beberapa nodus, diantaranya ada nodus sinoatrial, nodus



**Gambar 2. 1** Terjadinya Potensial Aksi[12]

atrioventrikular, berkas his, dan serabut purkinje, yang dapat dilihat pada Gambar 2.3.

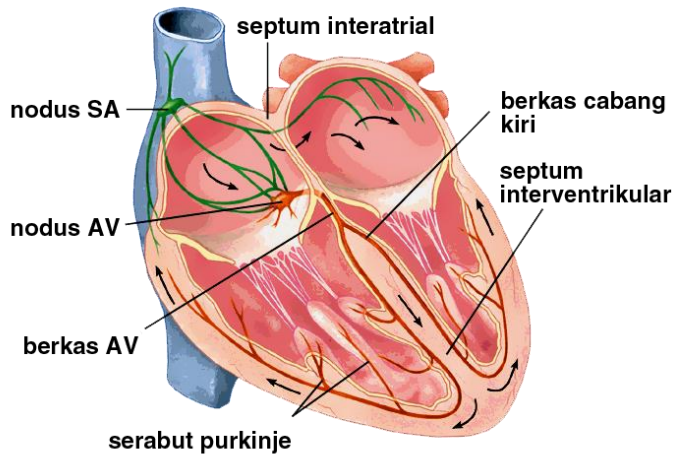
Nodus SA terletak pada perbatasan antara vena kava superior dengan atrium kanan. Sel dalam nodus SA mengeluarkan impuls 60-100 kali per menitnya secara teratur. Rangsangan menyebar ke seluruh atrium melalui 3 cabang intermodal yang terletak pada dinding atrium, dalam waktu 0,06 detik mencapai nodus AV.

Nodus AV sendiri terletak di atas sinus koronarius pada dinding posterior atrium kanan. Sel ini melepaskan impuls 40-60 kali per menit. Impuls ini lebih rendah dari nodus SA. Setelah impuls dari nodus SA ke nodus AV, impuls berhenti sebentar untuk menunggu ventrikel terisi darah dari atrium. Konduksi nodus ini relatif lambat yaitu 0,08-0,16 detik.

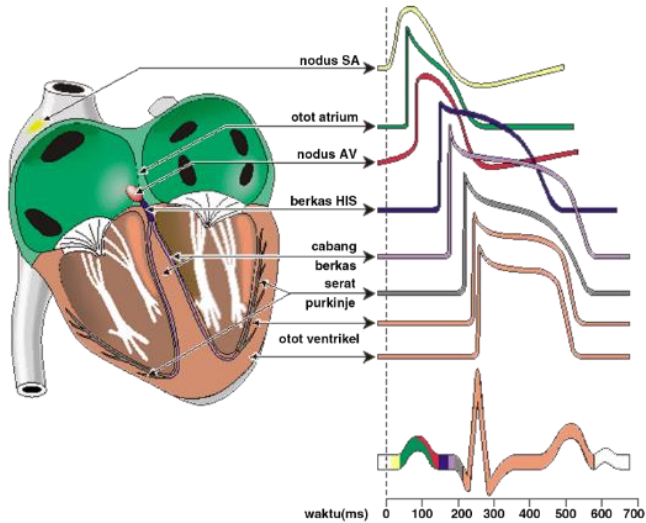
Berkas HIS adalah nama nodus yang menembus jaringan pemisah dinding otot/ miokardium antar atrium dan ventrikel, dan merambat pada septum ventrikel lalu bercabang dua menjadi berkas kanan dan kiri. Dari kedua cabang ini impuls bercabang menuju endokardium pada ventrikel kanan dan kiri, dan selanjutnya menyebar membentuk serabut serabut purkinje. Konduksi serabut his terjadi selama 0,03-0,05 detik. Serabut purkinje adalah serabut yang mampu mengeluarkan impuls dengan frekuensi 20-40 kali permenit. Serabut purkinje berisi ribuan fibril yang meliputi seluruh otot jantung [4],[5].

Pada jantung, dalam kondisi istirahat, permukaan luar sel otot jantung bermuatan positif, dan di dalam bermuatan negatif. Perbedaan muatan yang terjadi sekitar -90mV. Ketika ada rangsangan, stabilitas elektrik terganggu karena adanya pergeseran ion. Seperti dalam potensial aksi, rangsangan listrik menyebabkan masuknya ion  $\text{Na}^+$  dengan cepat ke dalam sel sehingga muatan dalam sel berubah positif dibanding luar sel. Proses ini berjalan bertahap dari ujung fibril sampai seluruh fibril. Proses perangsangan ini disebut depolarisasi. Setelah depolarisasi, terjadi pengembalian muatan ke keadaan semula yang disebut repolarisasi. Proses ini adalah potensial aksi yang terjadi di jantung.

Sinyal jantung dapat diklarifikasi menjadi sinyal PQRST. Setiap titiknya mewakili proses yang berbeda. Siklus dimulai dari sel nodus sinoatrial (SA) yang terletak di atrium kanan. Pada jantung orang dewasa yang sehat, simpul SA mengirimkan sinyal sebanyak 60 sampai 100 kali per menit. Dari nodus SA, sinyal merambat ke atrium kanan dan kiri. Hal ini menyebabkan atrium berkontraksi, sehingga darah dapat berpindah ke ruang jantung yang lebih rendah, yaitu ventrikel. Sinyal listrik yang bergerak melalui atrium ini tampak sebagai gelombang P.



**Gambar 2. 2** Anatomi Nodus Jantung[12]

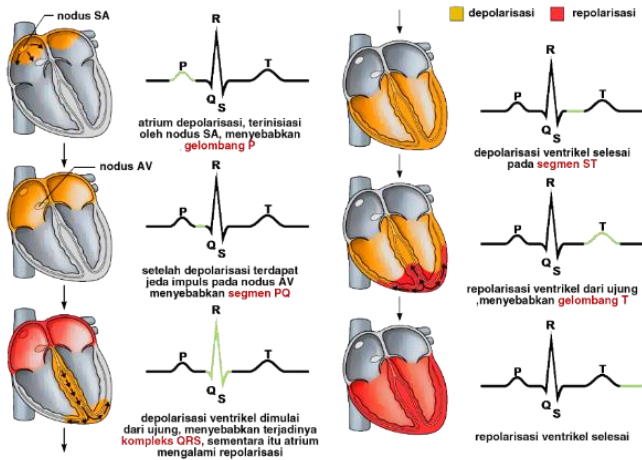


**Gambar 2. 3** Morfologi Sinyal Jantung[12]

Sinyal listrik merambat di antara atrium dan ventrikel melalui sekelompok sel yang disebut nodus atrioventrikular (AV). Sinyal melambat saat melewati nodus AV. Perlambatan ini memberi ventrikel cukup waktu untuk mengisi ruangan dengan darah. Karena melambat, sinyal yang tampak dari proses ini adalah garis datar antara akhir gelombang P dan awal gelombang Q. Garis datar ini berada di garis dasar, dan disebut juga dengan segmen PQ. Morfologi sinyal jantung dapat dilihat pada Gambar 2.4. dan pembentukan siklusnya dapat dilihat pada Gambar 2.4.

Sinyal listrik kemudian meninggalkan node AV menuju berkas HIS. Di sana, sinyal bercabang ke kanan dan kiri cabang berkas. Sinyal ini menyebar dengan cepat di seluruh ventrikel jantung, menyebabkan ventrikel untuk berkontraksi dan memompa darah ke paru-paru dan seluruh tubuh. Karena rangsangannya besar dan impuls, fenomena ini dalam sinyal jantung dapat dilihat pada titik Q,R, dan S. Segmen ini sering disebut sebagai segmen kompleks QRS

Setelah itu terjadi depolarisasi ventrikel, karena tidak ada sinyal yang berperan maka segmen yang terbentuk berupa garis datar yang disebut segmen ST. Ventrikel kemudian kembali ke kondisi normal mereka. Saat repolarisasi ini cenderung lama, karena terjadi proses pengisian ulang dari darah dari seluruh tubuh kembali ke jantung yang masuk melalui atrium. Proses ini ditampilkan sebagai gelombang T.



**Gambar 2. 4** Pembentukan Satu Siklus Sinyal Jantung[12]



### 2.1.3 Elektrokardiogram

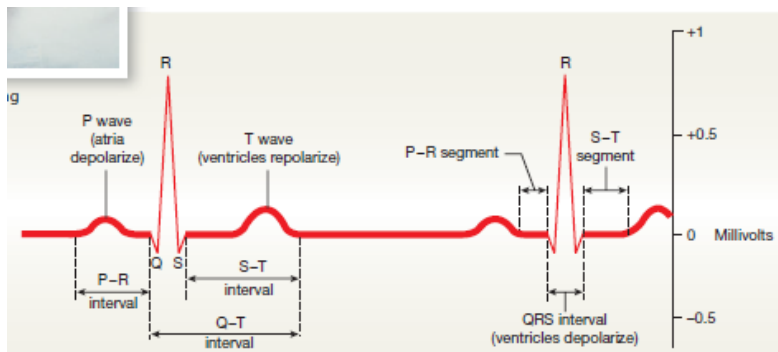
Momen elektrik di jantung cukup kuat untuk terdeteksi oleh elektroda di permukaan kulit. Perekaman momen listrik jantung disebut elektrokardiogram. Setiap kali jantung berdetak, gelombang depolarisasi menyebar melalui atrium, berhenti di AV node, kemudian menuju bawah interventricular septum ke puncak, memutar, dan menyebar melalui ventricular myocardium menuju bagian dasar jantung.

ECG mengintegrasikan informasi elektrik dari elektroda di tubuh. Dokter dapat menggunakan ECG untuk mendapatkan informasi jantung. Ketika terkena serangan jantung, ECG memberikan sinyal yang abnormal. ECG memiliki komponen :

- Gelombang P, muncul ketika depolarisasi atrium
- QRS complex, muncul ketika depolarisasi ventrikel, ventrikel mulai kontraksi setelah puncak R
- Gelombang T, mengindikasikan repolarisasi ventrikel

### 2.2. Elektrokardiograf

Sinyal kelistrikan yang dikeluarkan oleh jantung dapat dibaca dengan menggunakan elektrokardiograf. Elektrokardiogram atau EKG adalah alat kesehatan untuk mendeteksi sinyal potensial listrik pada jantung manusia dengan meletakkan elektroda di permukaan tubuh pasien. Contoh alat EKG konvensional dapat dilihat pada Gambar 2.7. Untuk merekam gelombang EKG diperlukan adanya perbedaan potensial dari titik-titik pengukuran pada permukaan tubuh.



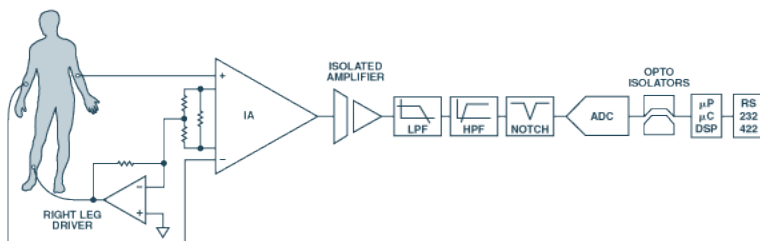
**Gambar 2. 5** Sinyal Elektrokardiogram Beserta Komponennya[7]

Prinsip kerja EKG cukup sederhana, tampak pada Gambar 2.6. Elektroda mengukur potensial tubuh dan ditangkap menjadi suatu sinyal. Sinyal tersebut kemudian dilewatkan ke dalam rangkaian penguat instrumentasi, lalu difilter untuk menghilangkan sinyal yang tidak diinginkan, dan dikonversikan ke data digital sebelum diproses lebih lanjut [10].

### 2.2.1. Sadapan EKG

Sadapan atau *lead* pada EKG merupakan konduktor yang dipasang di permukaan tubuh manusia untuk merekam aktivitas kelistrikan jantung dalam elektrokardiogram. Setiap sadapan menghasilkan perbedaan potensial yang menjadi sinyal aktivitas kelistrikan jantung sesuai dengan posisinya terhadap jantung. Secara umum, sadapan jantung dapat dibagi menjadi sadapan ekstrimitas dan prekordial. Karena kompleksitas otot jantung, maka untuk EKG diagnosa standar digunakan 12 sadapan dengan tujuan memberikan gambaran multidimensi atau menyeluruh terhadap sinyal jantung dan otot jantung yang menghasilkan sinyal tersebut. Namun untuk monitoring secara umum dapat digunakan elektroda dengan 3 sadapan ekstrimitas standar.

Sadapan ekstrimitas adalah sadapan yang dipasang di ekstrimitas tubuh yaitu lengan dan kaki. Sadapan ini memandang jantung dari arah frontal yang membentuk bidang vertikal. Berdasarkan potensial kelistrikannya, sadapan ekstrimitasnya dibagi menjadi dua, standar dan *augmented* atau tambahan. Sadapan ekstrimitas standar adalah sadapan yang bersifat bipolar, yakni merekam perbedaan potensial dua elektroda yang berbeda. Sadapan ini menggunakan metode



**Gambar 2. 6** Prinsip Kerja Elektrokardiograf[12]



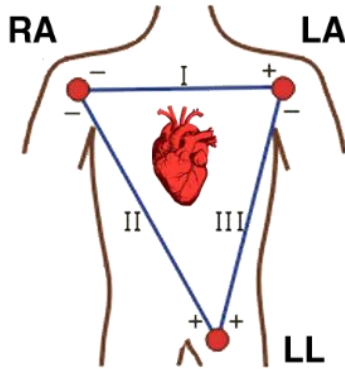
**Gambar 2. 7** EKG konvensional[12]

segitiga Einthoven dengan tiga buah sadapan ekstremitas standar, yaitu lead I, II, dan III, yang dapat dilihat pada Gambar 2.8. Metode ini menggambarkan kedua lengan dan kaki kiri membentuk segitiga yang mengelilingi jantung. Hukum Einthoven menyatakan bahwa jika potensial listrik dari setiap dua dari tiga elektroda ekstremitas bipolar diketahui, maka elektroda ketiga merupakan jumlah dari keduanya.

Sinyal EKG direkam dengan bantuan elektroda yang terpasang pada tiga titik tubuh, yaitu Right Arm, Left Arm, dan Left Leg sesuai dengan aturan segitiga Einthoven. Sadapan I merekam beda potensial antara tangan kanan (RA) yang bermuatan negatif dengan tangan kiri yang bermuatan positif dengan sudut orientasi  $0^\circ$ . Sadapan II merekam beda potensial antara tangan kanan yang bermuatan negatif dengan kaki kiri (LF) yang bermuatan positif dengan sudut orientasi  $60^\circ$ . Sadapan III merekam beda potensial antara tangan kiri (LA) yang bermuatan negatif dengan kaki kiri yang bermuatan positif dengan sudut orientasi  $120^\circ$ .

### **2.2.2. Penguat Instrumentasi**

Sinyal yang telah ditangkap dari elektroda dimasukkan ke dalam penguat instrumentasi. Penguat ini digunakan untuk memperkuat sinyal yang kecil dari elektroda, yang hanya berkisar milivolt. Agar dapat memproses sinyal tersebut sinyal harus diperbesar terlebih dahulu dengan amplifier. Rangkaian ini juga disebut *biopotential amplifier* karena menguatkan sinyal-sinyal dari dalam tubuh sementara tetap menjaga



**Gambar 2. 8** Segitiga Einthoven dan Sadapan Standar[12]

bentuk dan karakteristik sinyal asli dari dalam tubuh. Penguat instrumentasi terdiri dari rangkaian *double ended amplifier* dan rangkaian *differential amplifier*, selain itu terdapat rangkaian *driven left leg* untuk mengembalikan dan menyamakan tegangan ground pada kaki kiri, yang dapat dilihat pada Gambar 2.9. Analisa rangkaian dimulai dari perbedaan tegangan antara  $V_1$  dan  $V_2$  karena adanya *virtual ground* pada op-amp, maka dapat dituliskan dengan Persamaan 2.1.

$$V_1' = V_1 \quad , \quad V_2' = V_2 \quad (2.1)$$

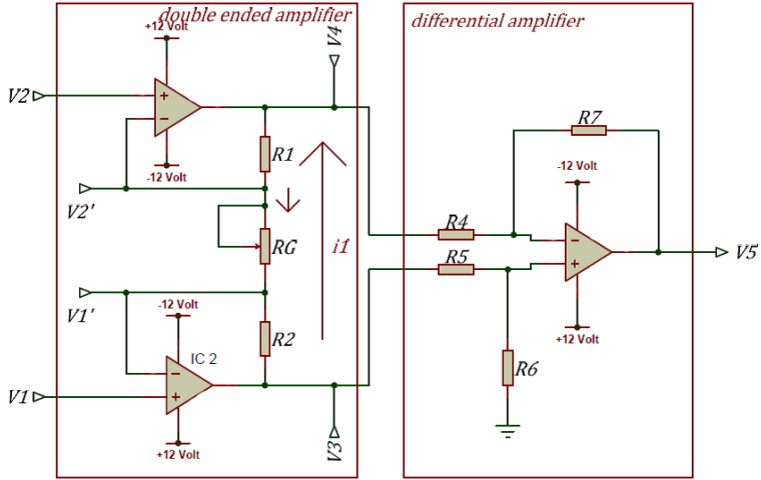
Beda potensial antara  $V_1$  dan  $V_2$  menimbulkan arus  $i_1$  yang mengalir antara  $V_3$  dan  $V_4$  dapat diturunkan pada Persamaan 2.2.

$$i_1 = \frac{V_1 - V_2}{R_G} \quad (2.2)$$

$$V_3 - V_4 = i_1 \cdot (R_1 + R_2 + R_G) \quad (2.3)$$

Yang dapat disubstitusikan menjadi

$$\frac{V_3 - V_4}{V_1 - V_2} = \frac{R_1 + R_2 + R_G}{R_G} \quad (2.4)$$



**Gambar 2. 9** Rangkaian Penguat Instrumentasi[12]

Karena  $R_1 = R_2 = R$  maka

$$\frac{V_3 - V_4}{V_1 - V_2} = \frac{2R}{R_G} + 1 \quad (2.5)$$

Keluaran rangkaian *double ended amplifier* menjadi masukan bagi penguat rangkaian diffrential, sehingga didapatkan Persamaan 2.6.

$$V_5 = V_{5|V_3} + V_{4|V_3} \quad (2.6)$$

$$V_{5|V_3} = \left( \frac{R_7}{R_4} + 1 \right) \cdot \left( \frac{R_6}{R_5 + R_6} V_3 \right) \quad (2.7)$$

$$V_{5|V_4} = - \left( \frac{R_7}{R_4} \cdot V_4 \right) \quad (2.8)$$

Jika disubstitusi didapatkan Persamaan 2.9 sebagai berikut

$$V_5 = \left[ \left( \frac{R_7}{R_4} + 1 \right) \cdot \left( \frac{R_6}{R_5 + R_6} \cdot V_3 \right) \right] - \left[ \frac{R_7}{R_4} \cdot V_4 \right] \quad (2.9)$$

Bila  $R_4 = R_5 = R_6 = R_7 = R$  maka didapatkan tegangan keluaran

$$V_5 = V_3 - V_4 \quad (2.10)$$

Bila disubstitusikan dengan rangkaian *double ended* dapat diperoleh hubungan input-ouput dari penguat instrumentasi 2.11.

$$V_{out} = \left( \frac{2R}{R_G} + 1 \right) \times (V_1 - V_2) \quad (2.11)$$

Persamaan penguatannya dengan resistor  $R_G$  diperoleh Persamaan 2.12.

$$G = \frac{49,4k\Omega}{R_G} + 1 \quad (2.12)$$

dimana  $R_G$  adalah resistor penguatan [14],[15],[16].

### 2.2.3. *High Pass Filter*

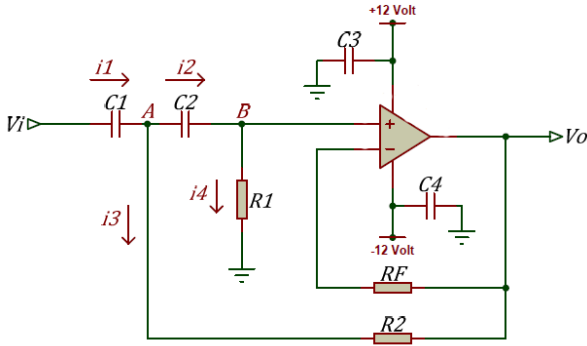
Sinyal masih harus dibersihkan dari sinyal DC yang berasal dari baterai. Oleh karena itu digunakan rangkaian *High Pass Filter* untuk meredam sinyal interferensi sinyal karena sinyal DC. Rangkaian ini memotong sinyal dengan frekuensi dibawah frekuensi *cut-off* ( $f_c$ ) dan melewatkan sinyal pada daerah frekuensi diatas frekuensi *cut-off*. Serupa dengan filter lainnya, saat frekuensi *cut-off*, penguatannya dapat dinyatakan sebagai Persamaan 2.13.

$$\left| \frac{V_o}{V_i} \right| = \frac{1}{\sqrt{2}} = 0.707 \quad (2.13)$$

Rangkaian pada Gambar 2.11 dirancang menggunakan topologi *Sallen-Key*, dengan orde 2, yang berarti penguatannya akan naik sebesar 40dB jika frekuensi naik sebesar 10 kali atau 1 dekade. Rangkaian ini bekerja dengan prinsip *buffer* dan *amplifier*. Berbeda dengan *low pass*, Pada *high pass* tegangan outputnya diumpun balikkan melalui resistor  $R_F$  ke masukan negatif op-amp dan melalui  $R_2$  ke masukan, baru dilewatkan kapasitor  $C_1$  dan  $C_2$ .

Rangkaian ini bekerja dengan perhitungan karena  $i_1 = i_2 + i_3$  dan  $i_1 = i + i_4 = i_4$  maka tegangan pada  $V_o$  sama dengan tegangan pada  $V_B$  sehingga dapat diturunkan persamaan berikut.





**Gambar 2. 10** Rangkaian HPF -40dB/dec[12]

$$i_2 = \frac{V_A - V_B}{1/j\omega \cdot C_2} = j\omega \cdot C_2 \cdot (V_A - V_o) \quad (2.14)$$

$$i_4 = \frac{V_B}{R_1} = \frac{V_o}{R_1} \quad (2.15)$$

Karena  $i_2 = i_4$  maka

$$(V_A - V_o) j\omega \cdot C_2 = \frac{V_o}{R_1} \quad (2.16)$$

$$V_A = V_o + \frac{V_o}{j\omega \cdot C_2 \cdot R_1} \quad (2.17)$$

$$i_1 = \frac{V_i - V_A}{1/j\omega \cdot C_1} = j\omega \cdot C_1 \cdot (V_i - V_A) \quad (2.18)$$

Bila disubstitusikan diperoleh

$$i_1 = \left( V_i - V_o - \frac{V_o}{j\omega \cdot C_2 \cdot R_1} \right) \cdot j\omega \cdot C_1 \quad (2.19)$$

$$i_4 = \frac{V_A - V_o}{R_2} \quad (2.20)$$

Karena  $i_1 = i_2 + i_3$  maka

$$\left(V_i - V_o - \frac{V_o}{j\omega.C_2.R_1}\right).j\omega.C_1 = (V_A - V_o).j\omega.C_2 + \frac{V_A - V_o}{R_2} \quad (2.21)$$

$$\left(V_i - V_o - \frac{V_o}{j\omega.C_2.R_1}\right).j\omega.C_1 = \left(V_o + \frac{V_o}{j\omega.C_2.R_1} - V_o\right).j\omega.C_2 + \frac{V_o + \frac{V_o}{j\omega.C_2.R_1} - V_o}{R_2} \quad (2.22)$$

$$\left(V_i - V_o - \frac{V_o}{j\omega.C_2.R_1}\right) = \frac{V_o}{j\omega.C_1.R_1} + \frac{V_o}{-\omega^2.C_1.C_2.R_1.R_2} \quad (2.23)$$

$$V_i = V_o + \frac{V_o}{j\omega.C_1.R_1} + \frac{V_o}{j\omega.C_2.R_1} - \frac{V_o}{\omega^2.C_1.C_2.R_1.R_2} \quad (2.24)$$

$$V_i = V_o \left\{ 1 - \frac{1}{\omega^2.C_1.C_2.R_1.R_2} + \frac{1}{j\omega.R_1} \left( \frac{1}{C_1} + \frac{1}{C_2} \right) \right\} \quad (2.25)$$

Sehingga persamaan akhirnya menjadi

$$\frac{V_o}{V_i} = \frac{1}{1 - \frac{1}{\omega^2.C_1.C_2.R_1.R_2} - j\frac{1}{\omega.R_1} \left( \frac{1}{C_1} + \frac{1}{C_2} \right)} \quad (2.26)$$

Pada saat  $\omega = \omega_c$  maka nilai penguatan dan sudut fasenya  $+90^\circ$ . Supaya menghasilkan nilai sudut fase tersebut maka nilai sudut pada penyebut sama dengan  $-90^\circ$  sehingga diperoleh persamaan 2.28.

$$1 - \frac{1}{\omega^2.C_1.C_2.R_1.R_2} = 0 \quad (2.27)$$

$$\omega_c = -\frac{1}{\sqrt{C_1.C_2.R_1.R_2}} \quad (2.28)$$

#### 2.2.4. Low Pass Filter

Setelah dikuatkan, sinyal masih harus dibersihkan dari *noise*. Oleh karena itu digunakan rangkaian *Low Pass Filter*. Rangkaian ini berfungsi untuk meredam interferensi gelombang RF / *radio frequency*. Rangkaian ini memotong sinyal dengan frekuensi diatas frekuensi *cut-off* ( $f_c$ ) dan melewatkan sinyal pada daerah frekuensi dibawah frekuensi *cut-*

off. Frekuensi *cut-off* adalah batas frekuensi antara frekuensi yang akan dihilangkan dan frekuensi yang akan dilewatkan. Saat frekuensi *cut-off*, penguatannya dapat dinyatakan dengan Persamaan 2.29

$$\left| \frac{V_o}{V_i} \right| = \frac{1}{\sqrt{2}} = 0.707 \quad (2.29)$$

Rangkaian pada Gambar 2.12. dirancang menggunakan topologi *Sallen-Key*, dengan orde 2, yang berarti penguatannya akan turun sebesar -40dB jika frekuensi naik sebesar 10 kali atau 1 dekade. Rangkaian ini bekerja dengan prinsip *buffer* dan *amplifier*. Tegangan output ditahan dan dibalikkan melalui resistor  $R_F$  ke masukan negatif op-amp dan diberi *feedback* oleh kapasitor  $C_2$ .

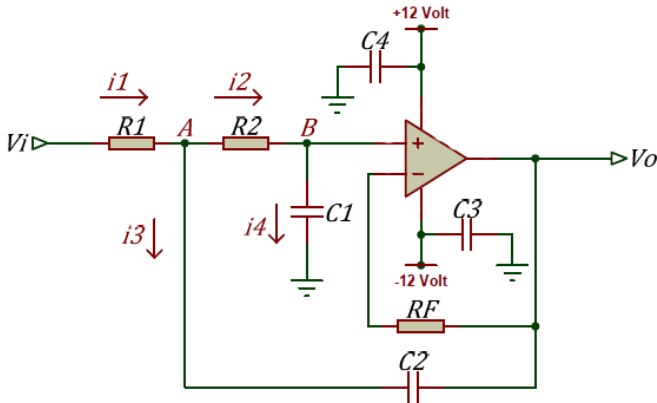
Rangkaian ini bekerja dengan penurunan persamaan sebagai berikut:

$$i_1 = i_2 + i_3 = i + i_4 = i_4 \quad (2.30)$$

$$i_4 = \frac{V_B}{1/j\omega C_1} = j\omega C_1 V_B = j\omega C_1 V_o \quad (2.31)$$

$$i_2 = \frac{V_A - V_B}{R_2} = \frac{V_A - V_o}{R_2} \quad (2.32)$$

Karena  $i_2 = i_4$  maka



**Gambar 2. 11** Rangkaian LPF -40dB/dec[12]

$$i_2 = i_4 = j\omega \cdot C_1 \cdot V_o \quad (2.33)$$

$$V_A = V_o + j\omega \cdot C_1 \cdot R_2 \cdot V_o \quad (2.34)$$

$$i_3 = \frac{V_A - V_o}{1/j\omega C_2} = (V_A - V_o) j\omega \cdot C_2 \quad (2.35)$$

Bila disubstitusikan diperoleh Persamaan 2.36.

$$i_3 = \omega^2 \cdot C_1 \cdot C_2 \cdot R_2 \cdot V_o \quad (2.36)$$

Karena  $i_1 = i_2 + i_3$  maka diperoleh Persamaan 2.37.

$$\frac{V_i - V_A}{R_1} = j\omega \cdot C_1 \cdot V_o + (-\omega^2 C_1 \cdot C_2 \cdot R_1 \cdot R_2 \cdot V_o) \quad (2.37)$$

$$V_i = V_o \{1 - \omega^2 \cdot C_1 \cdot C_2 \cdot R_1 \cdot R_2 + j\omega \cdot C_1 (R_1 + R_2)\} \quad (2.38)$$

Sehingga penguatan akhirnya diperoleh Persamaan 2.39.

$$\frac{V_o}{V_i} = \frac{1}{1 - \omega^2 \cdot C_1 \cdot C_2 \cdot R_1 \cdot R_2 + j\omega \cdot C_1 (R_1 + R_2)} \quad (2.39)$$

Bila  $\omega = \omega_c$  supaya menghasilkan sudut  $+90^\circ$  maka didapatkan Persamaan 2.41

$$1 - \omega_c^2 \cdot C_1 \cdot C_2 \cdot R_1 \cdot R_2 = 0 \quad (2.40)$$

$$\omega_c = \frac{1}{\sqrt{C_1 \cdot C_2 \cdot R_1 \cdot R_2}} \quad (2.41)$$

### 2.2.5. *Band Stop Filter*

Berbeda dengan rangkaian *low pass* dan *high pass*, rangkaian *band stop filter* berfungsi untuk meredap sinyal dengan frekuensi tertentu, yang memiliki lebar *bandwidth* tertentu pula. Pada rangkaian *band stop filter* ini, *bandwidth* dibatasi oleh frekuensi *cut-off* rendah (*low*) ( $\omega_{cL}$ ) dan frekuensi *cut-off* atas (*high*) ( $\omega_{cH}$ ). Sinyal yang akan dihilangkan atau dilemahkan yaitu sinyal yang ada diatas frekuensi *cut-off* rendah dan berada dibawah frekuensi *cut-off* atas, sedangkan sinyal yang akan dilewatkan yaitu sinyal yang berada dibawah frekuensi *cut-off* rendah dan

sinyal yang berada diatas frekuensi *cut-off* atas.Pada rangkaian ini LPF disebut juga sebaga *notch filter* karena digunakan untuk meredam sinyal jala jala yang berasal dari listrik rumah dengan frekuensi 50Hz dan bandwidth 10Hz. Rangkaian ini menggunakan topologi *Sallen-Key* dengan orde 2 seperti Gambar 2.12..

Agar dapat memenuhi persamaan penguatan  $\left| \frac{V_o}{V_i} \right| = \frac{1}{\sqrt{2}} = 0.707$

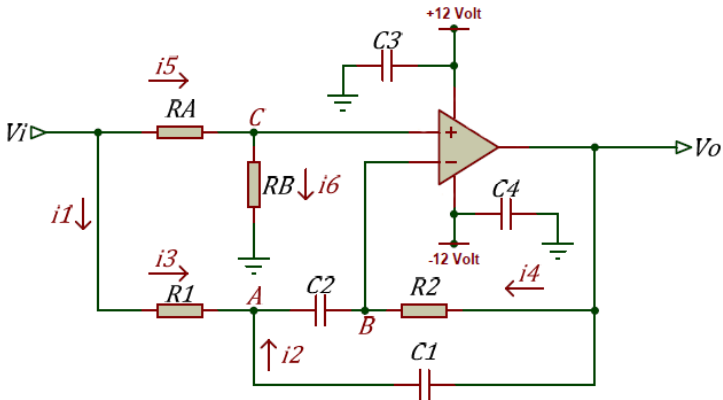
Maka digunakan frekuensi *cut-off* atas dan bawah. *Bandwidth* merupakan selisih antara frekuensi cut-off atas ( $\omega_{CH}$ ) dan frekuensi *cut-off* rendah ( $\omega_{CL}$ ) yang dapat ditulis sebagai  $B = \omega_{CH} - \omega_{CL}$  Sedangkan perbandingan antara lebar bidang frekuensi atau *bandwidth* dengan frekuensi resonansi disebut dengan faktor kualitas (Q), yang dapat ditulis dengan persamaan  $Q = \frac{\omega_c}{B}$  Untuk setiap arus dapat diperoleh persamaan

$$i_1 = \frac{V_i - V_A}{R_1} \quad (2.42)$$

$$i_2 = (V_o - V_A)j\omega \cdot C_1, i_3 = (V_A - V_B)j\omega \cdot C_2 \quad (2.43)$$

$$i_4 = \frac{V_o - V_B}{R_2}, i_5 = \frac{V_i - V_B}{R_2} \quad (2.44)$$

$$i_6 = \frac{V_B}{R_B} \quad (2.45)$$



**Gambar 2. 12** Rangkaian BPF -40dB[12]

$$V_B = V_C = \frac{R_B}{R_A + R_B} \cdot V_i \quad (2.46)$$

$$V_B \left( \frac{R_A + R_B}{R_A \cdot R_B} \right) = \frac{V_i}{R_A} \quad (2.47)$$

Berdasarkan hukum *kirchoff* 1 pada titik B,  $i_3 = i_4$  sehingga diperoleh Persamaan 2.48.

$$(V_A - V_B)j\omega \cdot C_2 + \frac{V_o - V_B}{R_2} = 0 \quad (2.48)$$

$$V_A = \frac{-V_o}{j\omega \cdot C_2 \cdot R_2} + V_B \left( 1 + \frac{1}{j\omega \cdot C_2 \cdot R_2} \right) \quad (2.49)$$

Berdasarkan Hukum *kirchoff* 1 pada titik A,  $i_1 + i_2 = i_3$  sehingga

$$\frac{V_i - V_A}{R_1} + (V_o - V_A)j\omega \cdot C_1 = (V_A - V_B)j\omega \cdot C_2 \quad (2.50)$$

$$\frac{V_o}{V_i} = \frac{-\frac{1}{R_1} + \frac{R_B}{R_A + R_B} \left( \frac{1}{R_1} + j\omega \cdot (C_1 + C_2) \right) + \frac{1}{j\omega \cdot C_2 \cdot R_1 \cdot R_2} + \left( \frac{C_1 + C_2}{C_1 \cdot C_2} \right) - j\omega \cdot C_2}{j\omega \cdot C_1 + \frac{1}{j\omega \cdot C_2 \cdot 2R_1 \cdot R_2} + \frac{C_1 + C_2}{C_1 \cdot C_2}} \quad (2.51)$$

$$\frac{V_o}{V_i} = \frac{-\frac{1}{R_1} + \frac{R_B}{R_A + R_B} \left( \frac{1}{R_1} + j\omega \cdot C + \frac{1}{j\omega \cdot C \cdot R_1 \cdot R_2} + \frac{2}{R_2} \right)}{j\omega \cdot C + \frac{1}{j\omega \cdot C \cdot R_1 \cdot R_2} + \frac{2}{R_2}} \quad (2.52)$$

$$Acl = \frac{V_o}{V_i} = \frac{1 + \frac{R_B}{R_A + R_B} \left( \left( 1 - \frac{2R_1}{R_2} \right) + j \left( \frac{1}{\omega \cdot C \cdot R_2} - \omega \cdot C \cdot R_1 \right) \right)}{-\frac{2R_1}{R_2} - j \left( \frac{1}{\omega \cdot C \cdot R_2} - \omega \cdot C \cdot R_1 \right)} \quad (2.53)$$

Frekuensi resonansi terjadi bila bagian imajiner sama dengan nol, atau

$$\frac{1}{\omega_r \cdot C \cdot R_2} - \omega_r \cdot C \cdot R_1 = 0 \quad (2.54)$$

$$\omega = \frac{1}{\sqrt{C \cdot R_1 \cdot R_2}} \quad (2.55)$$

Penguatan pada saat frekuensi input sama dengan frekuensi resonansi ( $A_r$ ) maka diperoleh Persamaan 2.57.

$$A_r = \frac{1 + \frac{R_B}{R_A + R_B} \left( -1 - \frac{2R_1}{R_2} \right)}{\frac{2R_1}{R_2}} \quad (2.56)$$

$$A_r = \frac{-R_2}{2R_1} + \frac{R_B}{R_A + R_B} \left( \frac{R_2 + 2R_1}{2R_1} \right) \quad (2.57)$$

Apabila penguatannya 0.707 maka dua buah frekuensi yaitu frekuensi *cut-off* bawah ( $\omega_{CL}$ ) dan frekuensi *cut-off* atas ( $\omega_{CH}$ ), lebar bidang frekuensi (*bandwidth*) dapat dituliskan sebagai persamaan 2.58.

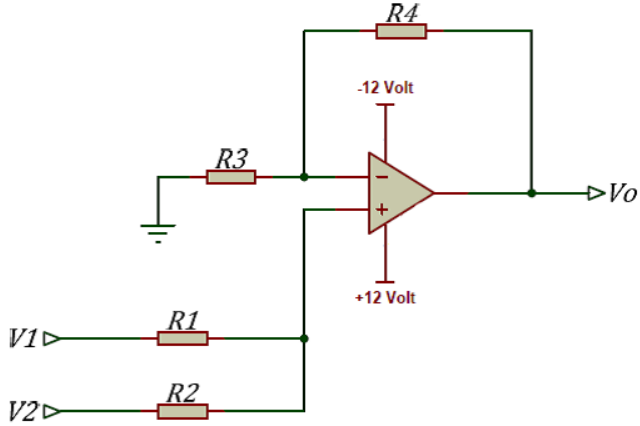
$$B = \frac{2}{R_2 \cdot C} \quad (2.58)$$

### 2.2.6. Rangkaian Penjumlah

Sebelum dilanjutkan ke pemrosesan sinyal digital, sinyal perlu dinaikkan agar dapat disampel dengan sempurna pada ADC. Oleh karena itu digunakan *non inverting adder* atau rangkaian penjumlah tak membalik. Rangkaian ini memanfaatkan tegangan referensi eksternal untuk menghasilkan outputan yang linier yang bergantung pada penjumlahan sinyal masukan dengan tegangan referensi tersebut. Secara teori terdapat dua macam rangkaian penjumlah, namun agar sinyal yang didapatkan sama dengan input, hanya tegangannya naik, maka rangkaian yang digunakan adalah rangkaian *non inverting amplifier* dengan penguatan satu kali seperti pada Gambar 2.13.

Pada rangkaian ini digunakan resistansi input yang sama,  $R_1 = R_2$  yang bertujuan untuk mendapatkan kestabilan dan akurasi penjumlahan sinyal inputan pada rangkaian, dan agar perbesarannya tidak memperhatikan resistansi masukan. Prinsip kerjanya mirip dengan *amplifier* dengan masukan tak membalik atau *non inverting*. Besar penguatan ( $A_v$ ) pada rangkaian ini dipengaruhi oleh tahanan *feedback*  $R_f$  dan tahanan *inverting*  $R_i$ .

Analisa rangkaian ini mirip dengan penjumlahan tegangan *non inverting amplifier*, apabila resistansi  $R_1 = R_2$  maka sesuai dengan teorema superposisi dapat digunakan Persamaan 2.59.



**Gambar 2. 13** Rangkaian Penjumlah [12]

$$V_o = V_{o1} + V_{o2} \quad (2. 59)$$

Dimana  $V_{o1}$  merupakan tegangan keluaran dari sumber 1 dan  $V_{o2}$  merupakan tegangan keluaran dari sumber 2.

$$V_{o1} = \left[ \frac{R_1}{R_1 + R_2} \cdot V_1 \right] \cdot \left[ \frac{R_4}{R_3} + 1 \right] \quad (2. 60)$$

$$V_{o2} = \left[ \frac{R_2}{R_1 + R_2} \cdot V_2 \right] \cdot \left[ \frac{R_4}{R_3} + 1 \right] \quad (2. 61)$$

Sehingga persamaan tegangan keluarannya adalah Persamaan 2.62.

$$V_o = \left( \frac{R_4}{R_3} + 1 \right) \cdot (V_1 + V_2) \quad (2. 62)$$

Untuk rangkaian ini digunakan V1 dari sinyal EKG, dan V2 dari tegangan referensi yang tetap.

### 2.3. Pemrosesan Sinyal Digital

Setelah sinyal analog sesuai dengan keinginan, sinyal dimasukkan ke dalam mikrokontroler atau mikroprosesor untuk diubah menjadi nilai digital agar dapat dikomputasi pada bagian ini terdiri dari pemrosesan sinyal analog ke digital, pengolahan multinode NRF24L01, komunikasi serial ke Raspberry Pi, pembuatan GUI. Pemrosesan sinyal



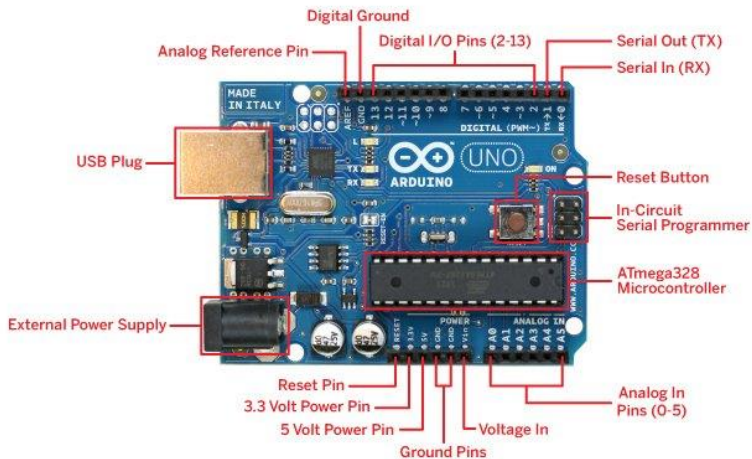
digital diperlukan agar sinyal dapat diidentifikasi dan disegmentasi sehingga output yang dihasilkan sesuai dengan yang diinginkan.

### 2.3.1. Arduino Uno

Arduino Uno R3 adalah papan pengembangan mikrokontroler yang berbasis chip ATmega328P. Arduino Uno memiliki 14 digital pin input / output (atau biasa ditulis I/O, dimana 14 pin diantaranya dapat digunakan sebagai output PWM antara lain pin 0 sampai 13), 6 pin input analog, menggunakan crystal 16 MHz antara lain pin A0 sampai A5, koneksi USB, jack listrik, header ICSP dan tombol reset. Hal tersebut adalah semua yang diperlukan untuk mendukung sebuah rangkaian mikrokontroler.

Spesifikasi Arduino Uno :

Mikrokontroler	: ATmega328
Operasi Tegangan	: 5 Volt
Input Tegangan	: 7-12 Volt
Pin I/O Digital	: 14
Pin Analog	: 6
Arus DC tiap pin I/O	: 50 mA
Arus DC ketika 3.3V	: 50 mA
Memori Flash	: 32 KB



**Gambar 2. 14** Arduino Uno R3

SRAM : 2 KB  
EEPROM : 1 KB  
Kecepatan clock : 16 MHz

Arduino Uno memiliki 6 channel, 10-bit analog to digital converter. Yang berarti ADC tersebut akan memetakan tegangan input antara 0 dan 5 volt ke nilai integer antara 0 dan 1023. Hal tersebut menghasilkan pembacaan dengan resolusi antara 5 volts/1024 units, atau 0.0049 volts (4.9mV) per unit. Membutuhkan waktu sekitar 100 mikrosekon (0.0001 s) untuk membaca analog input, sehingga maksimal pembacaan data sekitar 10.000 data per detik.

### **2.3.2. NRF24L01**

Modul Wireless nRF24L01 adalah sebuah modul komunikasi jarak jauh yang memanfaatkan pita gelombang RF 2.4GHz ISM (Industrial, Scientific and Medical). Modul ini menggunakan antarmuka SPI untuk berkomunikasi. Tegangan kerja dari modul ini adalah 5V DC.

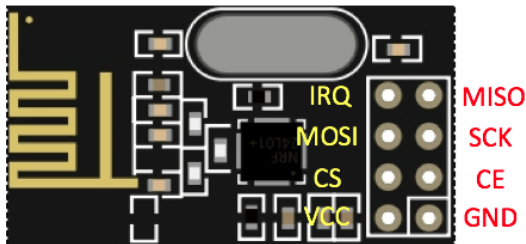
NRF24L01 memiliki baseband logic Enhanced ShockBurst™ hardware protocol accelerator yang support “high-speed SPI interface for the application controller”. nRF24L01 memiliki true ULP solution, yang memungkinkan daya tahan baterai berbulan-bulan hingga bertahun-tahun.

Modul ini memiliki 8 buah pin, diantaranya :

1. VCC(3.3 Volt DC)
2. GND
3. CE (Chip Enable)
4. CSN (Chip Select)
5. MOSI (Master Output Slave Input)
6. MISO ( Master Input Slave Output)
8. IRQ

Fitur yang terdapat di NRF24L01 :

1. Beroperasi pada pita ISM 2.4 GHz
2. Data rate hingga 2Mbps
3. Ultra low power
4. Penanganan paket data otomatis
5. Penanganan transaksi paket otomatis.



**Gambar 2. 15** NRF24L01 [18]

### 2.3.3. Komunikasi SPI

Serial Peripheral Interface ( SPI ) merupakan salah satu mode komunikasi serial synchronous berkecepatan tinggi. Komunikasi SPI membutuhkan 3 jalur yaitu MOSI, MISO, dan SCK. Melalui komunikasi ini data dapat saling dikirimkan baik antara mikrokontroller maupun antara mikrokontroller dengan peripheral lain di luar mikrokontroller. Penjelasan 3 jalur utama dari SPI adalah sebagai berikut :

- MOSI : Master Output Slave Input Artinya jika dikonfigurasi sebagai master maka pin MOSI sebagai output tetapi jika dikonfigurasi sebagai slave maka pin MOSI sebagai input.
- MISO : Master Input Slave Output Artinya jika dikonfigurasi sebagai master maka pin MISO sebagai input tetapi jika dikonfigurasi sebagai slave maka pin MISO sebagai output.
- CLK : Clock Jika dikonfigurasi sebagai master maka pin CLK berlaku sebagai output tetapi jika dikonfigurasi sebagai slave maka pin CLK berlaku sebagai input.

### 2.3.4. Raspberry Pi

Raspberry Pi (atau Raspi) adalah sebuah komputer papan tunggal (SBC) berukuran kartu kredit yang dihubungkan ke TV (via HDMI) dan keyboard. Sebagai IoT (Internet of Things), seperti layaknya sebuah desktop, PC kecil ini mampu digunakan untuk menjalankan spreadsheet, pengolah kata dan permainan, terutama untuk memainkan video definisi tinggi.

Desain Raspi didasarkan seputar SoC (System-on-a-chip) Broadcom BCM2835, yang telah menanamkan prosesor ARM1176JZF-S dengan 700 MHz, VideoCore IV GPU, dan 256 Megabyte RAM (model B). Penyimpanan data didisain tidak untuk menggunakan hard disk atau

solid-state drive, melainkan mengandalkan kartu SD (SD memory card) untuk booting dan penyimpanan jangka panjang. Raspi utamanya menjalankan sistem operasi berbasis kernel Linux.

Sistem operasi utama Raspberry Pi menggunakan Debian GNU/Linux, mengemas Iceweasel, kaligrafi Suite dan bahasa pemrograman Python. Sejumlah distro lainnya, termasuk distro buatan Indonesia BlankOn Linux yang juga telah menyediakan versi arsitektur ARM dan bisa dijalankan di Raspi. Hardware Raspberry Pi tidak memiliki real-time clock, sehingga OS harus memanfaatkan timer jaringan server sebagai pengganti. Namun komputer yang mudah dikembangkan ini dapat ditambahkan dengan fungsi real-time (seperti DS1307) dan banyak lainnya, melalui saluran GPIO (General-purpose input/output) via antarmuka I<sup>2</sup>C (Inter-Integrated Circuit). Gambar Raspberry Pi dapat dilihat pada Gambar 2.16.

### 2.3.5. Pengolahan Sinyal Digital

Pada pengolahan sinyal jantung, pemrosesan sinyal digital difokuskan untuk memisahkan segmen sinyal jantung dan mengekstraksi parameter tertentu dari segmentasi tersebut. Contohnya segmentasi dapat digunakan untuk memperoleh segmen kompleks QRS, dan mengekstraksi parameter R-R interval. Ada tiga teorema yang dapat digunakan, yakni



**Gambar 2. 16** Raspberry Pi

dengan *band pass filter*, filter gabungan, dan *windowed search*.

Disini hanya dijelaskan teorema kedua karena tugas akhir ini menggunakan filter gabungan tersebut. Filter gabungan ini dapat digunakan karena *Band Pass Filter* juga dapat diperoleh dari gabungan antara *Low Pass Filter* dan *High Pass Filter* dimana nilai filter ditentukan dengan dua frekuensi *cutoff* yang berbeda sehingga membentuk *band pass*. Nilai kedua filter ini dibedakan berdasarkan frekuensi segmen jantung. Apabila digunakan orde yang berbeda dapat diperoleh bentuk *bandpass* yang tidak simetris sehingga dapat diatur untuk mengikuti *bandwidth* segmen sinyal jantung tertentu. Apabila digunakan metode HPF dan LPF maka dapat digunakan HPF dengan orde 2 yang dapat dirumuskan sebagai Persamaan 2.63. [12].

$$y(n) = \frac{(A.y(n-1) - B.y(n-2) + C.x(n) - D.x(n-1) + E.x(n-2))}{F} \quad (2.63)$$

$$\begin{aligned} A &= \frac{8}{T_s^2} - 2\omega c^2 \\ B &= \frac{4}{T_s^2} - \frac{2\sqrt{2}\omega c}{T_s} + \omega c^2 \\ C &= \frac{4}{T_s^2} \\ D &= \frac{8}{T_s^2} \\ E &= \frac{4}{T_s^2} \\ F &= \frac{4}{T_s^2} + \frac{2\sqrt{2}\omega c}{T_s} + \omega c^2 \end{aligned}$$

Sedangkan untuk LPF digunakan orde 2 dengan Persamaan 2.64

$$y(n) = \frac{(A.y(n-1) - B.y(n-2) + C.x(n) + D.x(n-1) + E.x(n-2))}{F} \quad (2.64)$$

$$\begin{aligned} A &= \frac{8}{T_s^2} - 2\omega c^2 \\ B &= \frac{4}{T_s^2} - \frac{2\sqrt{2}\omega c}{T_s} + \omega c^2 \\ C &= \omega c^2 \\ D &= 2\omega c^2 \\ E &= \omega c^2 \end{aligned}$$

$$F = \frac{4}{T_s^2} + \frac{2\sqrt{2}\omega c}{T_s} + \omega c^2$$

Dengan nilai  $\omega c = 2\pi \cdot fc$  (2. 65)

*fc = Frekuensi cut off*

*Ts = time sampling*

*Halaman ini sengaja dikosongkan*

### BAB III

## PERANCANGAN SISTEM

Pada bab perancangan sistem akan dipaparkan mengenai desain keseluruhan sistem dan spesifikasinya. Perancangan dari Pemantauan Electrocardiogram Secara Online Menggunakan Raspberry Pi Untuk Pemeriksaan EKG Jarak Jauh ini terdiri dari empat bagian, yaitu pada elektrokardiogram, ADC, NRF24L01, Raspberry Pi. Pada perancangannya sendiri dapat dibagi menjadi dua, yakni perancangan perangkat keras elektrokardiogram, dan perancangan perangkat lunak. Diagram blok sistem secara keseluruhan dapat dilihat pada Gambar 3.2.

Dari diagram blok dapat dilihat bahwa sinyal diproses dengan tiga tahap. Tahap pertama pada instrumentasi EKG, sinyal jantung diperoleh dari tubuh melalui elektroda, dan diproses secara analog pada EKG, dan dihasilkan sinyal analog yang siap untuk dimasukkan ADC. Pada ADC, sinyal diubah menjadi digital. Sinyal yang didapat dari beberapa client atau transmitter kemudian dikirimkan menggunakan modul NRF24L01. ADC dari beberapa Arduino Uno dikirim secara multinode. Multinode disini menggunakan beberapa transmitter dan 1 receiver. Setelah receiver menerima data, dikirimkan secara serial ke Raspberry Pi untuk pembuatan GUI atau *interface*. Sebagai sumber daya sistem elektrokardiogram digunakan dua baterai tegangan +9V dan -9V

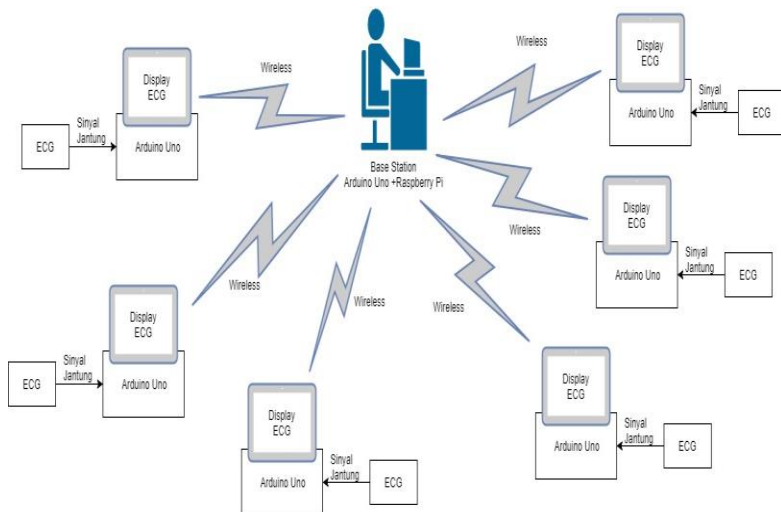
#### 2.4. Perancangan Perangkat Elektrokardiogram

Perangkat keras yang digunakan adalah elektrokardiogram yang secara umum terdiri dari tiga bagian, yakni penguat, penyaring, dan penambah. Pada rangkaian EKG tujuan utama rangkaian adalah merubah sinyal dari sinyal jantung yang memiliki amplitudo kecil dan memiliki banyak *noise* menjadi sinyal jantung analog yang bersih dari *noise* dan memiliki amplitudo yang cukup besar dengan tegangan minimum di atas nol agar dapat dimasukkan dan dibaca dengan benar oleh *analog-to-digital converter*.



**Gambar 3. 1** Diagram Blok Rangkaian EKG[12]





**Gambar 3. 2** Diagram Blok Sistem Keseluruhan

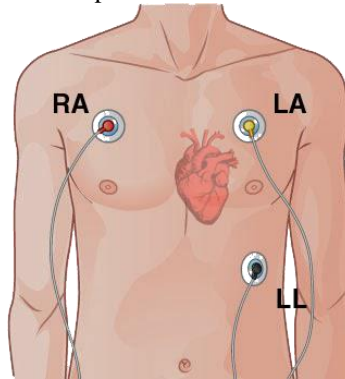
Sinyal jantung memiliki amplitudo maksimal mencapai 2mV dengan rentang frekuensi 0.05 hingga 100 Hz. Oleh karena itu terdapat banyak tahapan dalam perancangan EKG yang diantaranya adalah *Instrumentasi Amplifier*, *High Pass Filter (HPF)*, *Low Pass Filter (LPF)*, *Band Stop Filter (BSF)*, dan rangkaian penjumlah tak membalik. Diagram blok proses rangkaian EKG dan penempatan sadapan elektroda einthoven dapat dilihat pada Gambar 3.1. dan Gambar 3.3., dan interpretasinya menjadi skematik rangkaian dapat dilihat pada lampiran.

### 3.1.1. Perancangan Penguat Instrumentasi

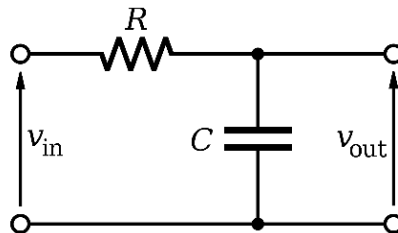
Pertama-tama sinyal dari tubuh ditangkap oleh tiga buah elektroda berdasarkan teorema Einthoven dengan 3 *Lead*, yang diletakkan pada *Left Arm / LA*, *Right Arm / RA*, dan *Left Leg / LL*. Ketiga elektroda itu mewakili elektroda ekstermitas, sehingga tidak harus diletakkan di ujung tangan, namun lebih baik bila diletakkan di dada penggunaanya seperti pada Gambar 3.3.

Setelah ditangkap elektroda, sinyal dilanjutkan ke penguat instrumentasi. Penguatan ini bertujuan untuk membesarkan sinyal yang awalnya hanya memiliki amplitudo maksimal dengan orde milivolt

hingga menjadi sinyal dengan orde volt. Oleh karena itu digunakan AD620 dengan skematik seperti Gambar 3.5.



**Gambar 3. 3** Penempatan Elektroda[12]



**Gambar 3. 4** Filter *Low Pass* Pasif

Penguat yang digunakan sebagai rangkaian instrumentasi dibagi menjadi empat bagian. Pada bagian pertama dari sinyal yang masuk ke tubuh diberi rangkaian filter pasif. Hal ini dilakukan untuk mengurangi efek redaman yang disebabkan oleh *gain-bandwidth product*. Teorema tersebut menyatakan bahwa penguatan dari rangkaian instrumentasi akan berkurang terhadap tinggi frekuensi masukannya. Hasil dari perkalian antara *gain* atau penguatan dikali dengan *bandwidth* atau pita frekuensi sinyal selalu sama, sehingga semakin tinggi frekuensi masukan maka penguatan akan semakin berkurang.

Filter sederhana ini memiliki frekuensi *cut-off* yang cukup tinggi, yakni 160 Hz. Karena digunakan kapasitor dengan nilai 100nF, maka nilai resistansi dapat diperoleh dengan persamaan 3.1..

$$f_c = \frac{1}{2\pi.R.C} \quad (3. 1)$$

$$160 \text{ Hz} = \frac{1}{2 \cdot \pi \cdot 100nF \cdot R}$$

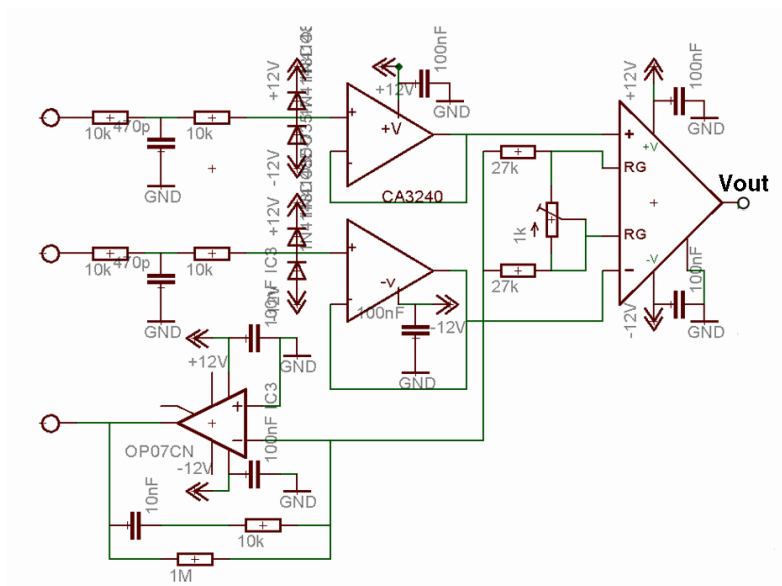
$$R \cong 10k\Omega$$

Setelah dilewatkan filter terdapat rangkaian proteksi yang akan dijelaskan pada sub bab 3.1.6. mengenai rangkaian pengamanan. Rangkaian kemudian dimasukkan dalam rangkaian penguatan utama yang terdiri dari 3 buah *op-amp*, yakni CA3240, AD620, dan OP07. Pertama tama, IC CA3240 yang berisi dua buah *operational amplifier* digunakan sebagai rangkaian *buffer* atau penyangga agar tidak ada arus balik. Oleh karena itu digunakan CA3240 yang memiliki spesifikasi rangkaian yang impedansi masukannya tinggi  $Z_{IN}$  hingga  $1.5T\Omega$  dan arus masukannya sangat rendah, hingga  $10pA$  pada  $\pm 15V$ .

Setelah itu sinyal positif dan negatif dimasukkan dalam rangkaian AD620. IC ini dipilih pun karena memiliki karakteristik impedansi input yang besar, yakni sekitar  $10M\Omega$ . Selain itu, rangkaian ini memiliki *common mode rejection ratio* atau CMRR yang tinggi yaitu  $100dB$  pada saat perbesarannya 10 kali, Sehingga IC ini dipilih sebagai penguat instrumen. CMRR sendiri adalah rasio dari penguatan rangkaian pada penguatan *common-mode*, yang diukur dalam satuan desibel positif. Pada amplifier instrumentasi, seperti AD620 ini, nilai CMRR biasanya tinggi, CMRR disini menjadi ukuran perubahan tegangan output ketika kedua input diubah dengan perubahan yang sama. Untuk CMRR optimal, terminal referensi harus diberi impedansi rendah, dan perbedaan nilai kapasitansi dan resistansi antara input harus sekecil mungkin. Amplitudo sinyal EKG berada dalam orde mV, sedangkan keluaran yang diinginkan adlah sinyal dengan orde V, maka dirancang penguatan  $\pm 1000$  kali. Besar penguatan diatur melalui resistor variabel yang dipasang pada tahanan RG, sehingga nilai RG dapat dikalkulasi dengan Persamaan 3.2.

$$1000 = \frac{49,4k\Omega}{RG} + 1 \quad (3.2)$$

$$RG \cong 50\Omega$$



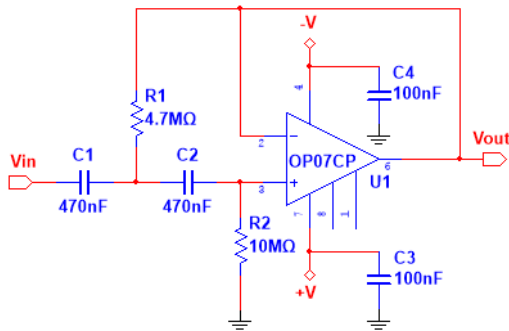
**Gambar 3. 5** Rancangan Rangkaian Instrumentasi

Walaupun sudah digunakan IC AD620, masih terdapat *noise* yang signifikan. Karena sinyal masukannya diperbesar, maka walaupun *noise* sudah difilter, tetapi akan ikut teramplifikasi juga. Oleh karena itu digunakan rangkaian *drive right leg* yang menggunakan IC OP07. Rangkaian *drive left leg* sendiri diperoleh berdasarkan rekomendasi datasheet AD620. Rangkaian ini bekerja dengan prinsip *buffer* dengan menggunakan resistor *feedback* yang bernilai  $1\text{M}\Omega$ , dan dua resistor input yang bernilai  $27\text{k}\Omega$ . Terdapat pula kapasitor untuk menjaga kestabilan rangkaian. Rangkaian *drive left leg* ini memiliki masukan setengah dari selisih tegangan antara AD620. Rangkaian ini digunakan untuk memberi masukan tegangan agar terjadi keseimbangan *ground* antara tubuh dengan rangkaian. Keluaran rangkaian ini dihubungkan dengan kaki atau *Right Leg* sebagai sumber referensi. Sehingga rangkaian instrumentasi dapat dibuat seperti Gambar 3.5.

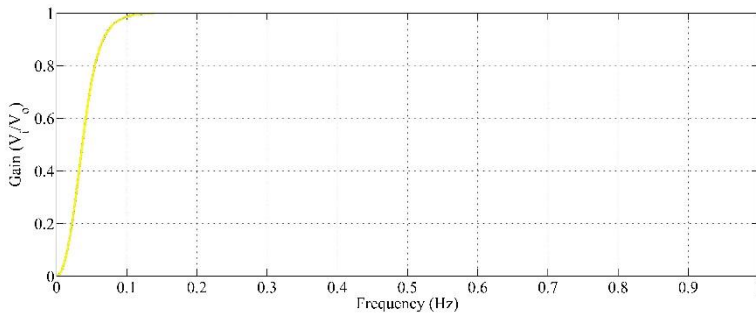
### 3.1.2. Perancangan *High Pass Filter*

Setelah melewati penguatan rangkaian masih memiliki *noise*. Noise tersebut dihilangkan dengan cara dilewatkan rangkaian *low pass filter*. Rangkaian ini digunakan untuk meredam interferensi sinyal listrik otot. Selain itu masih tercampur pula tegangan DC yang terbawa dengan rangkaian. Oleh karena itu sinyal dilewatkan *High Pass Filter* untuk menghilangkan interferensi tersebut. Perancangan rangkaian HPF ini dapat dilihat Gambar 3.6 dan respon frekuensinya pada Gambar 4.7.

*High Pass Filter* ini didesain dengan orde 2 dan 1 tahapan. Agar tidak memotong sinyal jantung, HPF dibuat dengan frekuensi *cut-off* sekecil mungkin, yakni pada 0,05 Hz. Sehingga dapat dihitung dengan Persamaan 3.3, apabila



**Gambar 3. 6** Rancangan Rangkaian HPF



**Gambar 3. 7** Respon Frekuensi HPF

$$R_2 = 4.7M\Omega$$

$$R_1 = \frac{1}{2}R_2 = 10M\Omega$$

Dapat dihitung nilai kapasitansi  $C_1$  dan  $C_2$  apabila  $C_1 = C_2$  maka

$$\omega_c = - \frac{1}{\sqrt{C_1 \cdot C_2 \cdot R_1 \cdot R_2}} \quad (3.3)$$

$$0,05 = \frac{1}{2\pi\sqrt{470nF \cdot 470nF \cdot R_1 \cdot R_2}}$$

### 3.1.3. Perancangan *Low Pass Filter*

Sinyal EKG yang telah diberi *high pass filter* masih memiliki *noise* yang berasal dari interferensi RF dan mencegah *anti aliasing* ketika disampling yang terjadi karena lingkungan sekitar. Sama dengan HPF, *Low Pass Filter* didesain dengan orde 2, menggunakan topologi *Sallen-key*, namun rangkaian ini menggunakan 3 tahapan agar dapat meredam dengan sempurna. *Bandwidth* frekuensi monitoring EKG dari sinyal jantung berkisar hingga 40 Hz. Frekuensi *cut-off* dari *low pass filter* ini ditetapkan pada 40 Hz. Sehingga dapat dianalisa apabila

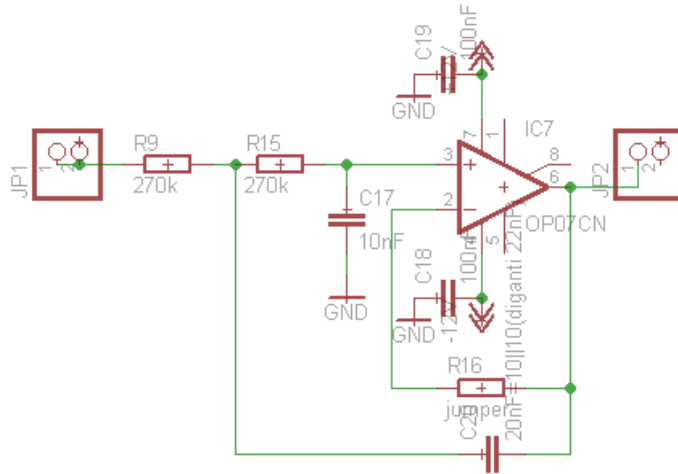
$$C_1 = 10nF$$

$$C_2 = 2 \times C_1 = 20nF \approx 22nF$$

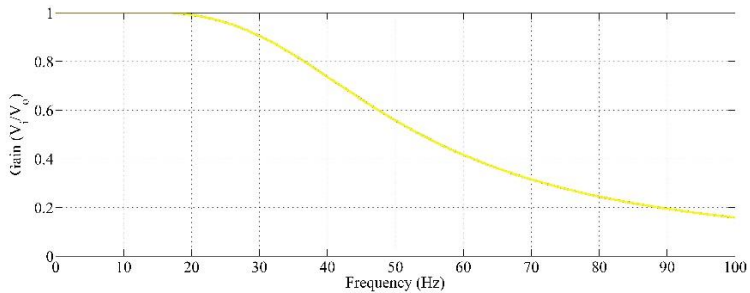
Maka dapat dirancang rangkaian dengan persamaan 3.4.

$$f_c = \frac{1}{2\pi\sqrt{C_1 \cdot C_2 \cdot R_1 \cdot R_2}} \quad (3.4)$$

$$40 \text{ Hz} = \frac{1}{2\pi\sqrt{10nF \cdot 22nF \cdot R_1 \cdot R_2}}$$



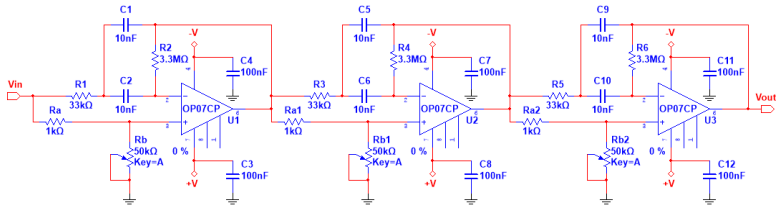
**Gambar 3. 8** Rancangan Rangkaian LPF



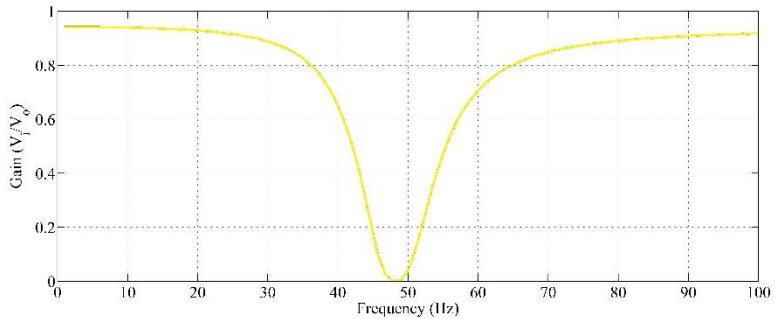
**Gambar 3. 9** Respon Frekuensi LPF

$$R = R_1 = R_2 = 270\text{k}\Omega$$

Sehingga rangkaian dapat dirancang seperti Gambar 3.10 dengan respon frekuensi seperti Gambar 3.11.



**Gambar 3. 10** Rancangan Rangkaian BPF



**Gambar 3. 11** Respon Frekuensi Rangkaian BPF

### 3.1.4. Perancangan *Band Stop Filter*

Pada praktiknya, sinyal yang sudah terpotong *high pass filter* dan *low pass filter* tersebut masih belum bersih dari *noise*. Masih ada interferensi sinyal jala jala dari lingkungan sekitar. Sinyal ini masuk ke rangkaian melalui komponen komponen yang terbuka sehingga dapat menangkap frekuensi seperti layaknya antena. Oleh karena itu, *Band Stop Filter* atau yang biasa disebut rangkaian *Notch* ini memiliki frekuensi *cut-off* 50Hz. Sama dengan LPF, *Low Pass Filter* didesain dengan orde 2, menggunakan topologi *Sallen-key* dengan 3 tahapan agar dapat meredam dengan sempurna, seperti Gambar 3.10 dengan respon frekuensi seperti Gambar 3.11. Pada perancangannya, apabila frekuensi yang diinginkan adalah 50 Hz dengan lebar *bandwidth* 10Hz, maka dapat dihitung dengan persamaan berikut,

$$C_1 = C_2 = 10nF$$

$$\omega_r = 2\pi f = 2 \cdot 3,14 \cdot 50 = 314$$



$$\omega_B = 2\pi f = 2 \cdot 3,14 \cdot 10 = 62,8$$

$$Q = \frac{\omega_r}{\omega_B} = \frac{3,14}{62,8} = 5$$

$$R_2 = \frac{2}{\omega_B \cdot C} = \frac{2}{62,8 \cdot 10 \cdot 10^{-9}} = \frac{2 \cdot 10^9}{628} = 3.3\text{M}\Omega$$

$$R_1 = \frac{R_2}{4 \cdot Q^2} = \frac{3,2 \cdot 10^6}{4 \cdot 5^2} = 32 \cdot 10^3 = 33\text{K}\Omega$$

### 3.1.5. Perancangan Rangkaian Penjumlah

Tahap selanjutnya adalah rangkaian penjumlah atau *Adder*. Sinyal EKG memiliki rentang tegangan dengan rata rata negatif. Sinyal ini tidak dapat dibaca oleh *analog to digital converter* karena ADC internal ARM memiliki range pembacaan 0V hingga 3V. Rangkaian ini digunakan untuk menaikkan *baseline* sinyal EKG yang memiliki range -1 volt hingga 1 volt. Keluaran rangkaian ini adalah rangkaian dengan nilai minimum di atas 0 volt. Oleh karena itu digunakan rangkaian penjumlah tak membalik atau *non inverting adder* seperti Gambar 3.12.

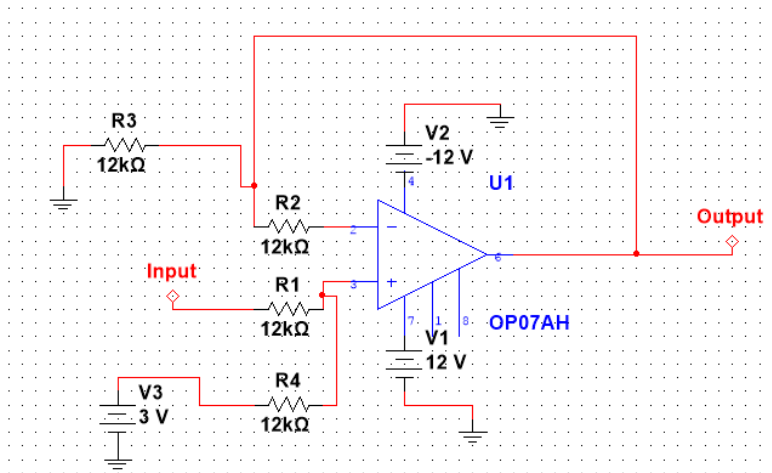
Untuk penjumlahan sinyal digunakan rangkaian OP07 dengan sumber tegangan penambah dari baterai 3 volt.

$$V_{out} = V_1 + V_2$$

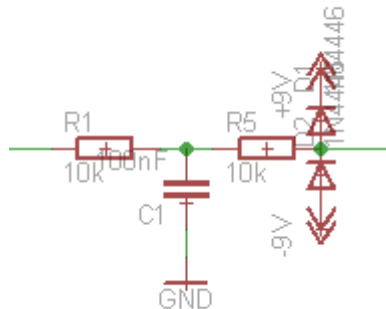
Dimana  $V_1$  adalah sinyal,  $V_2$  adalah tegangan *offset* yang ditambahkan, dan  $V_{out}$  adalah tegangan keluarannya yang menjadi masukan ADC.

### 3.1.6. Perancangan Rangkaian Proteksi

Selain rangkaian utama pemrosesan sinyal, rangkaian yang perlu diperhatikan adalah rangkaian pengaman. Rangkaian ini berfungsi untuk memastikan keamanan rangkaian dan keamanan penggunaanya. Untuk itu diperlukan beberapa pengaman yang dibuat dari dioda yang dipasang seperti Gambar 3.13..



**Gambar 3. 12** Rancangan Rangkaian Penjumlah



**Gambar 3. 13** Rangkaian Proteksi RC dan Dioda

Pada bagian penguat instrumentasi telah dijelaskan bahwa rangkaian awal menggunakan filter pasif RC. Selain digunakan untuk *prefilter* sinyal, rangkaian ini juga dapat menjadi rangkaian pengaman karena melindungi rangkaian dari arus dan frekuensi tinggi.

Setelah itu terdapat rangkaian proteksi. Rangkaian ini terdiri dari dua resistor dan empat dioda, dimana resistor digunakan untuk membatasi arus, sedangkan dioda berfungsi sebagai pembatas tegangan. Apabila tegangan masukan berada diantara +9 hingga -9 Volt, maka sinyal akan dilewatkan. Apabila sinyal masukan berada di atas +9 Volt, maka dioda

positif menjadi *forward bias* dan dioda negatif menjadi *reverse bias* sehingga arus mengalir ke dioda positif. Sebaliknya, apabila sinyal masukan berada di bawah -9 Volt, maka dioda positif akan menjadi *reverse bias* dan dioda negatif menjadi *forward bias* sehingga arus mengalir ke dioda negatif. Rangkaian ini juga digunakan sebagai proteksi EKG apabila dalam kondisi tertentu jantung pasien perlu dipicu menggunakan alat pacu jantung atau *defibrillator*. Alat *defibrillator* ini memiliki tegangan yang sangat tinggi yang dapat merusak instrumentasi EKG. Rangkaian proteksi ini juga dapat digunakan untuk melindungi instrumentasi EKG dari pengaruh *defibrillator* tersebut.

### **3.2. Perancangan Sistem Piranti Lunak**

Setelah diproses secara analog, sinyal diproses secara digital di mikrokontroller Arduino, modul NRF24L01, dan Raspberry Pi. Sinyal analog di konversi menjadi sinyal digital oleh ADC Arduino Uno. Sinyal digital dari masing masing *client* dikirim menggunakan NRF24L01. Setiap *client* mengirim data sinyal digital ke *server* atau *receiver* NRF24L01. Data yang diterima server NRF24L01 dikirimkan ke Raspberry Pi secara komunikasi serial USB. Pengolahan GUI atau *interface* diprogram menggunakan bahasa C++, menggunakan *software Qt Creator*.

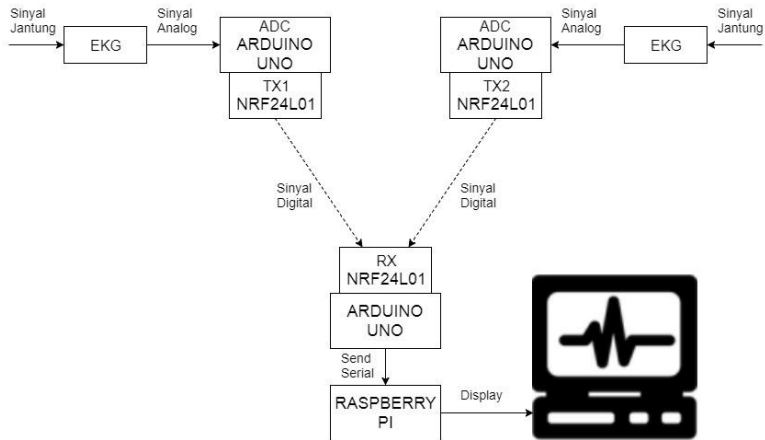
#### **3.2.1. Perancangan Analog To Digital Converter**

Raspberry Pi tidak memiliki pin ADC. Maka dari itu, dibutuhkan rangkaian ADC dari luar untuk membantu Raspberry Pi dalam membaca sensor. Dalam tugas akhir digunakan ADC Arduino Uno. Fungsi dari ADC tersebut ialah mengkonversi data analog dari hasil pembacaan ekg menjadi data digital. Sehingga Raspberry Pi dapat membaca ekg dengan output ADC.

Dalam melakukan proses pembacaan ADC, output ekg dimasukkan ke pin analog A0 Arduino Uno dan ground ekg ke pin ground ekg.

#### **3.2.2. Perancangan Transmisi Data Multinode NRF24L01**

Pada subbab ini dirancang sistem yang dapat mengirim data EKG tiap node untuk dikirim ke satu *receiver* atau *base station*. Pada tugas akhir ini hanya merancang 2 node sebagai transmitter dan 1 receiver untuk dibuat rancangan awal.



**Gambar 3. 14** Diagram Blok Sistem Piranti Lunak

### 3.2.2.1 Perancangan Program Transmitter 1

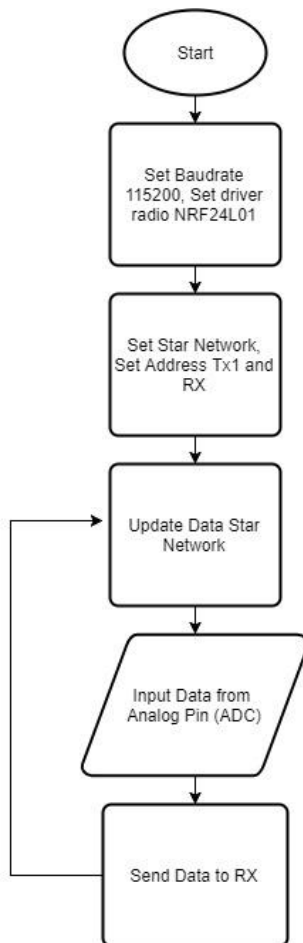
Program dimulai dengan mengimport library SPI, radio NRF24L01, dan network. Kemudian mengatur baudrate 115200 dan radio driver NRF24L01. Untuk multinode, dimulai dengan menginisialisasi star network dan mengatur address Transmitter 1 dan Receiver.

Loop dimulai dengan update data star network atau multinode. Data diambil dari pembacaan ADC melalui pin analog, kemudian data ADC dikirim ke receiver. Loop ini selalu berputar ketika data ADC yang baru didapatkan. Flowchart program transmitter dapat dilihat pada Gambar 3.15.

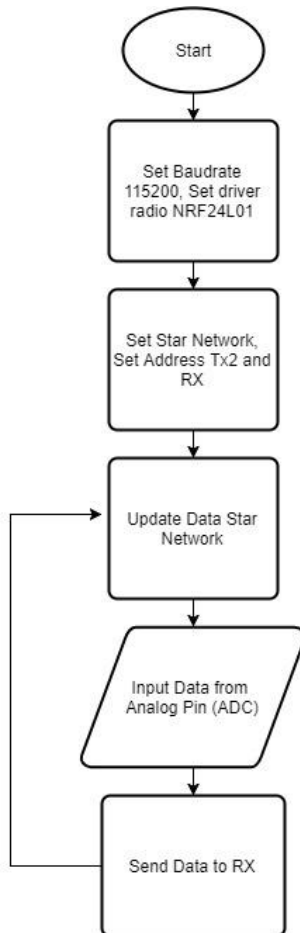
### 3.2.2.2 Perancangan Program Transmitter 2

Perancangan program transmitter 2 hampir sama dengan transmitter 1, yang membedakan adalah pengaturan address transmitter. Program dimulai dengan mengimport library SPI, radio NRF24L01, dan network. Kemudian mengatur baudrate 115200 dan radio driver NRF24L01. Untuk multinode, dimulai dengan menginisialisasi star network dan mengatur address Transmitter 2 dan Receiver.

Loop dimulai dengan update data star network atau multinode. Data diambil dari pembacaan ADC melalui pin analog, kemudian data ADC dikirim ke receiver. Loop ini selalu berputar ketika data ADC yang baru didapatkan.



**Gambar 3. 15** Flowchart program transmitter 1



**Gambar 3. 16** Flowchart program transmitter 2

### 3.2.2.3 Program Receiver

Program dimulai dengan mengimport library SPI, radio NRF24L01, dan network. Kemudian mengatur baudrate 115200 dan radio driver NRF24L01. Untuk multinode, dimulai dengan menginisialisasi star network dan mengatur address Receiver.

Loop dimulai dengan update data star network atau multinode. Data diambil dari pembacaan masing masing node atau masing masing transmitter, kemudian data masing masing node di serial println dengan format (data node1, data node2). Format data ini digunakan agar software GUI QT Creator dapat membedakan data tiap node dan memplotnya. Loop ini selalu berputar ketika data baru tiap node didapatkan. Flowchart program receiver dapat dilihat pada gambar 3.17.

### **3.2.3. Raspberry Pi**

Dalam perancangan sistem ini, Raspberry Pi berfungsi sebagai pengolah GUI atau *interface* ekg dari semua transmitter. Pengolahan data yang didapat dari server NRF24L01 dikirim secara serial USB dari Arduino Uno ke Raspberry Pi. Menggunakan *software Qt Creator* sebagai pengolah GUI.

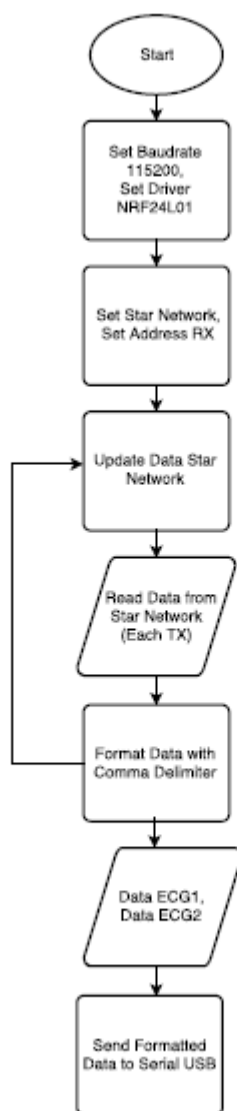
Diperlukan suatu penginisialisasi serial communication antara Arduino Uno dan Raspberry Pi. Terdapat beberapa bagian, yaitu, PortName, BaudRate, DataBits, dan Parity.

Selama serial komunikasi bekerja, data digital dari kedua transmitter masuk ke Raspberry Pi. Karena data yang masuk merupakan gabungan dua data transmitter, perlu program untuk pemisahan kedua data.

Setelah pemisahan data antara data ekg 1 dan ekg 2, berikut program untuk memplot data yang diperoleh secara real time dengan sumbu Y data yang diterima dan sumbu X waktu saat ini. Flowchart program plot pada Qt Creator di Raspberry Pi dapat dilihat pada gambar 3.18.

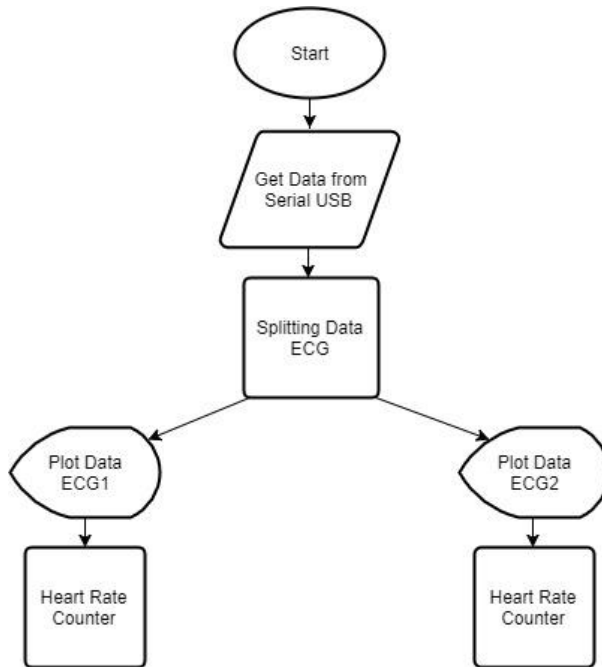
### **3.2.4. Perancangan Deteksi Segmentasi**

Setelah diplotl, sinyal dilewatkan pada band pass filter. Berbeda dengan pemrosesan sinyal analog, band pass filter pada pemrosesan sinyal digital digunakan untuk mengisolasi sinyal pada segmen tertentu. Sinyal yang dikuatkan adalah sinyal pada kompleks QRS yang memiliki frekuensi berbeda dengan sinyal pada segmen lainnya, dengan konfigurasi berbeda untuk segmentasi.



**Gambar 3. 17** Flowchart program receiver





**Gambar 3. 18** Flowchart program GUI *QT Creator*

Untuk deteksi kompleks QRS, dapat digunakan metode filter gabungan antara *Low Pass Filter* dan *High Pass Filter* dimana nilai filter ditentukan dengan dua frekuensi *cutoff* yang berbeda sehingga membentuk *bandwidth* dari *band pass filter*. . Pada LPF digunakan orde 2 yang dapat dirumuskan sebagai Persamaan 3.5

$$y(n) = \frac{(A.y(n-1) - B.y(n-2) + C.x(n) - D.x(n-1) + E.x(n-2))}{F} \quad (3.5)$$

Dimana :

$$A = \frac{8}{Ts^2} - 2\omega c^2$$

$$B = \frac{4}{Ts^2} - \frac{2\sqrt{2}\omega c}{Ts} + \omega c^2$$

$$C = \frac{4}{Ts^2}$$

$$D = \frac{8}{Ts^2}$$

$$E = \frac{4}{Ts^2}$$

$$F = \frac{4}{Ts^2} + \frac{2\sqrt{2}\omega c}{Ts} + \omega c^2$$

Untuk segmen QRS digunakan *low pass filter* dengan spesifikasi berikut.  
 $Ts = 0.02s$  dan  $\omega c = 2\pi f c = 2\pi(20Hz)$

*High Pass Filter* yang digunakan menggunakan filter dengan orde 3 yang dapat dirumuskan dalam Persamaan 3.6

$$y(n) = \frac{(A.y(n-1) - B.y(n-2) + C.x(n) - D.x(n-1) + E.x(n-2))}{F} \quad (3.6)$$

Dimana :

$$A = \frac{8}{Ts^2} - 2\omega c^2$$

$$B = \frac{4}{Ts^2} - \frac{2\sqrt{2}\omega c}{Ts} + \omega c^2$$

$$C = \frac{4}{Ts^2}$$

$$D = \frac{8}{Ts^2}$$

$$E = \frac{4}{Ts^2}$$

$$F = \frac{4}{Ts^2} + \frac{2\sqrt{2}\omega c}{Ts} + \omega c^2$$

Untuk segmen QRS digunakan *high pass filter* dengan spesifikasi berikut.

$$Ts = 0.02s \text{ dan } \omega c = 2\pi f c = 2\pi(10Hz)$$

### 3.2.5. Perancangan Komputasi Data

Setelah dilewatkan *band pass filter* maka sinyal sudah dapat dideteksi dan disegmentasi. Setelah didapatkan segmen yang diperlukan,

sinyal dikomputasi untuk memperoleh parameter irama jantung dan indikasi serangan jantung. Komputasi data dilakukan menggunakan modifikasi dari algoritma *Pan-Tompkins*, yaitu *squaring*, *moving window*, dan *thresholding*.

Tahapan pertama adalah *squaring* atau kuadrat. Pada tahapan ini sinyal yang besarnya lebih dari 1 dikuatkan dan sinyal yang besarnya kurang dari 1 diredam, sehingga sinyal QRS yang telah diisolasi menjadi semakin besar dan berorientasi positif. Persamaan yang digunakan adalah Persamaan 3.7 berikut:

$$y[n] = x[n] \cdot x[n] = (x[n])^2 \quad (3.7)$$

Dan pada program di Qt Creator di Raspberry dapat dituliskan untuk setiap data:

```
y [n] = qPow ( x [n] , 2 ) ;
```

Tahapan selanjutnya adalah *moving window integral* atau *moving average filter*. Tahapan ini digunakan untuk memperhalus sinyal dan menghilangkan *noise* dan distorsi. Tahapan ini merupakan cara untuk mendeteksi QRS menggunakan jumlah sampel tertentu. Pemilihan nilai  $M$  sampel harus diuji untuk menghindari masuknya gelombang T pada tahapan ini, namun kondisi yang baik adalah saat sampel  $M$  mendekati nilai sebelum gelombang T. Persamaan rumus *moving window integral* dapat dituliskan sebagai Persamaan 3.8 berikut.

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i - j] \quad (3.8)$$

Jika dipilih nilai *window*  $M = 10$

$$y[n] = \frac{1}{10} (x[n] + x[n - 1] + x[n - 2] + \dots + x[n - 9]) \quad (3.9)$$

Sehingga dapat dituliskan listing program sebagai berikut, untuk setiap datanya:

```
for (k=0; k<M; k++) {
    y [n] = y [n] + ( x [n - (M-k) ] ) / M;
}
```

Tahap terakhir adalah *thresholding*. Tahapan ini digunakan sebagai deteksi segmentasi. Jika sinyal lebih besar dari *threshold* maka

segmen terdeteksi. Jika tidak terdeteksi maka logika bernilai *false* atau 0. Sehingga dapat ditulis Persamaan 3.10.

$$y[n] = \begin{cases} 1, x[n] > threshold \\ 0, x[n] < threshold \end{cases} \quad (3.10)$$

Untuk mendeteksi tepi naik dicari nilai kenaikan dimana nilai masukkan sebelumnya lebih kecil dari ambang batas *threshold* atau dapat dituliskan pada Persamaan 3.11.

$$y[n] = \begin{cases} 1, x[n] > threshold \cap x[n-1] < threshold \\ 0, lainnya \end{cases} \quad (3.11)$$

Sehingga pada program dapat dituliskan dengan *conditional IF* atau percabangan dengan sintaks sebagai berikut.

```
if (x[0]>threshold){
    if(risingedge == true){
        komputasidata;
    }
    risingedge = 0;
}
else{
    risingedge = true;
}
```

Untuk segmentasi kompleks QRS dihitung jarak *R-R interval* dapat dihitung dengan menghitung jarak tepi naik tersebut. Setelah jarak diketahui, jarak yang masih memiliki satuan sampel tersebut diubah menjadi nilai *heart rate*. Setelah itu dapat diperoleh pula variabilitas detak jantung yang merupakan nilai perubahan detak jantung yang dapat dihitung berdasarkan Persamaan 3.12.

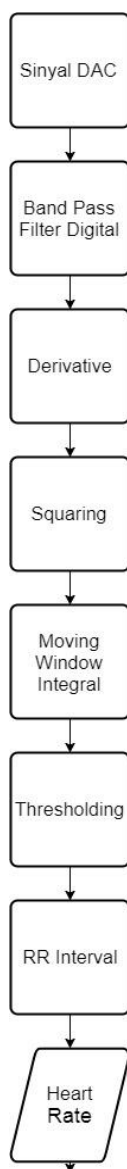
$$HR = \frac{f_{sampler}}{RR_{interval}} \text{ beat per detik} \quad (3.12)$$

$$HR = \frac{f_{sampler} \times 60}{RR_{interval}} \text{ bpm} = HR = \frac{60000}{RR_{interval}} \text{ bpm} \quad (3.13)$$

Sehingga dapat dituliskan sintaks program sebagai berikut.

```
if (x[0]>threshold){
    if(risingedge == true){
        HR=60000/RRint;
        RRint=0;
        sprintf(strs, "%d-", HR);}
    }
}
RRint++;
```

Keseluruhan algoritma untuk menghitung heart rate dapat dilihat pada Gambar 3.19.



**Gambar 3. 19** Flowchart perhitungan heart rate

*Halaman ini sengaja dikosongkan*

## BAB IV

### PENGUJIAN DAN ANALISIS

Setelah dilakukan perancangan maka akan dilakukan tahapan percobaan atau pengujian untuk menguji apakah sistem yang dirancang sudah sesuai dengan teori atau sesuai dengan kinerja yang diharapkan. Pengujian dilakukan satu per satu untuk tiap tahapannya. Pengujian dibagi dua, yakni pengujian perangkat elektrokardiogram dan pengujian sistem piranti lunak.

#### 4.1. Pengujian Perangkat Elektrokardiogram

Pengujian dan analisa perangkat elektrokardiogram dibagi menjadi beberapa bagian, yaitu pengujian penguat instrumentasi, *Low Pass Filter*, *High Pass Filter*, *Band Stop Filter*, rangkaian penjumlah, dan rangkaian pengaman. Khusus untuk filter *high pass filter* dan *band pass filter* digunakan 3 *stage* yang dipasang dengan *cascade* untuk memperhalus filter. Perancangan alat dilakukan di Semua rangkaian tersebut diujikan agar karakteristiknya sesuai dengan spesifikasi awalnya.

##### 4.1.1. Pengujian Instrumentation Amplifier IC AD620

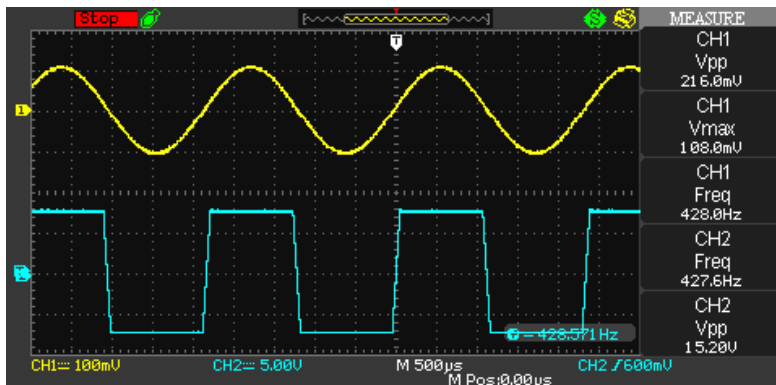
Pengujian penguat instrumentasi dilakukan dengan memasukkan sinyal masukan sinusoidal dengan amplitudo kecil, kemudian sinyal dikuatkan dengan rangkaian instrumentasi sehingga diperoleh besar penguatan rangkaian. Pengujian dilakukan dengan memberikan sinyal sinus dari *function generator* dengan amplitudo  $\pm 220\text{mV}$  dan frekuensi  $\pm 428\text{ Hz}$ . Sinyal ini diuji dengan dimasukkan pada input positif pada penguat instrumen, sedangkan input negatif dihubungkan dengan *ground*. RG yang merupakan tahanan yang diatur dengan tahanan resistor variabel dengan nilai  $80\Omega$ .

Pada rangkaian ini melalui *differential mode* dapat diperoleh penguatan rangkaian instrumentasi. Hasil pengujian penguatan dapat dilihat di Gambar 4. 1

Dengan *channel 2* sebagai keluaran dan *channel 1* sebagai masukan. Dengan RG bernilai  $80\Omega$  , diperoleh perhitungan teori sebagai berikut:

$$Av = \frac{49,4k\Omega}{80\Omega} = 617.5 \text{ kali}$$





**Gambar 4. 1** Hasil Pengujian *Instrumentation Amplifier*

Pada Gambar 4.1, *channel 2* terjadi saturasi dikarenakan sifat opamp. Yaitu tegangan output maksimum mempunyai range antara 1-2V dibawah tegangan sumber atau supply. Karena supply menggunakan baterai 9V dan -9V, sehingga saturasi terjadi di sekitar 16Vpp.

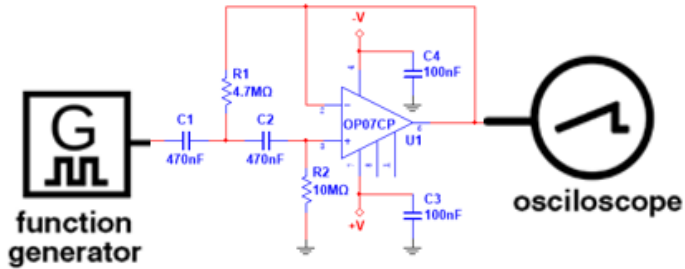
#### 4.1.2. Pengujian *HighPass Filter*

Pengujian *High Pass Filter* dilakukan dengan memberi sinyal masukkan sinyal sinus dengan amplitudo yang sama pada frekuensi berbeda beda. Frekuensi tegangan ini terletak di sekitar frekuensi *cutoff* sehingga dapat diketahui respon frekuensi dari rangkaian. *High Pass Filter* lebih sulit untuk diamati karena kendala function generator di frekuensi rendah. Gambar rangkaian dapat dilihat pada Gambar 4.2 dan hasil pengujiannya pada Tabel 4. 1. Respon frekuensi pada Gambar 4.3 dan Tampilan saat frekuensi terkecil function generator pada Gambar 4.4.

*High Pass Filter* idealnya dirancang mempunyai frekuensi *cutoff* 0.05Hz, maka secara teori pada saat frekuensi input sama dengan frekuensi *cutoff* nya diharapkan penguatannya sebesar 0.707. berdasarkan pengujian yang dilakukan,

- frekuensi pada saat penguatan 0.707 tidak diketahui karena keterbatasan function generator untuk menghasilkan sinyal dengan frekuensi dibawah 0.0798Hz.
- Penguatan maksimum pada daerah *passband* adalah 1 kali

- Penguatan sama dengan nol kali saat frekuensi 0Hz.



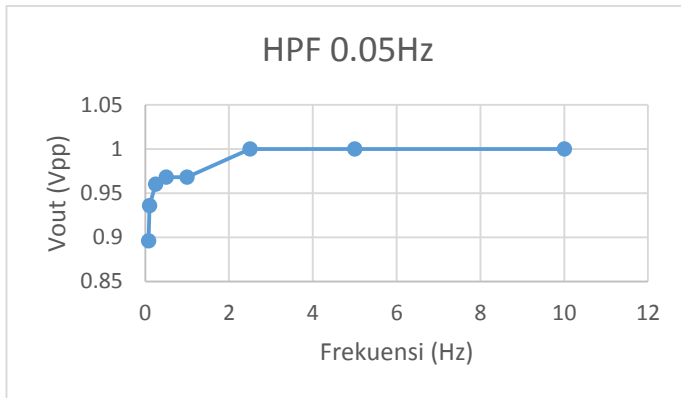
**Gambar 4. 2** Rangkaian Pengujian *High Pass Filter*

**Tabel 4. 1** Pengujian *High Pass Filter*

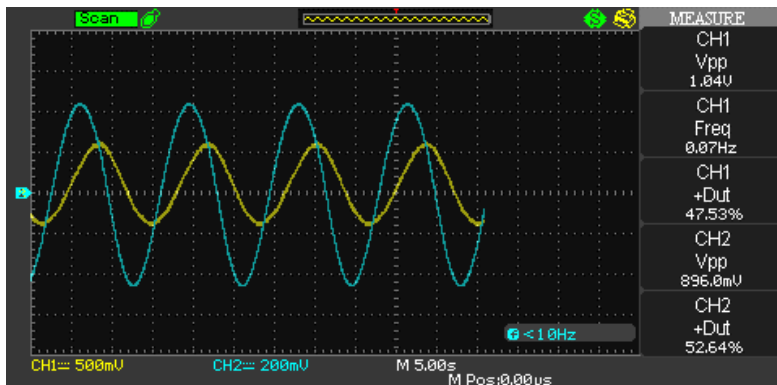
frekuensi (Hz)	Vpp input (Volt)	Vpp output (Volt)
10	1	1
5	1	1
2.5	1	1
1	1	0.968
0,5	1	0.968
0,25	1	0.96
0,1	1	0.936
0,0798	1	0.896

#### 4.1.3. Pengujian *Low Pass Filter*

Seperti filter sebelumnya, pengujian *Low Pass Filter* dilakukan dengan memberi sinyal masukkan sinyal sinus dengan amplitudo yang sama pada frekuensi berbeda. Skematik rangkaian dapat dilihat pada Gambar 4.5, hasil pengujian dapat dilihat pada Tabel 4.2.



**Gambar 4. 3** Grafik Pengujian *High Pass Filter*

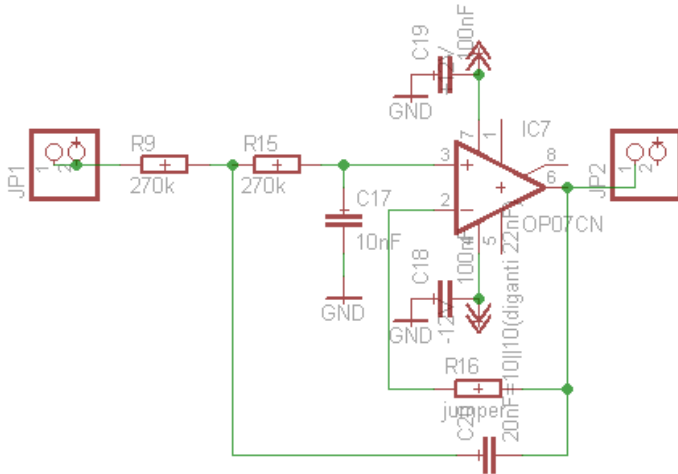


**Gambar 4. 4** Pengukuran HPF saat frekuensi terendah di function generator (0.0798 Hz)

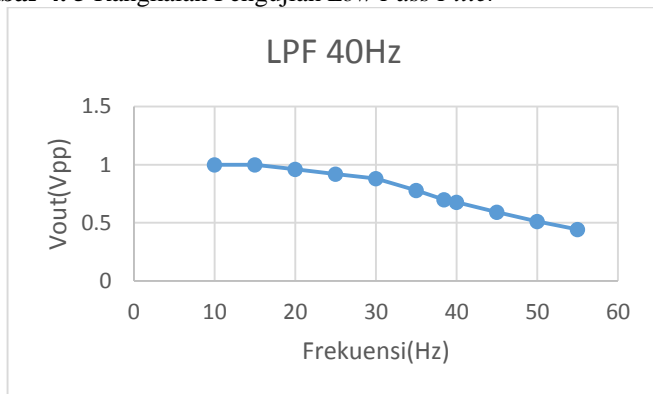
Frekuensi tegangan ini terletak di sekitar frekuensi *cutoff* sehingga dapat diketahui respon frekuensi dari rangkaian pada Gambar 4.6.

*Low Pass Filter* idealnya dirancang mempunyai frekuensi *cutoff* 40Hz, maka secara teori pada saat frekuensi tegangan input sama dengan frekuensi tegangan *cutoff* nya, maka penguatan yang terjadi sebesar 0.707. berdasarkan pengujian yang dilakukan,

- frekuensi pada saat penguatan 0.707 adalah 38.46 Hz.
- Penguatan maksimum pada daerah *passband* adalah 1 kali.
- Kesalahan frekuensi cutoff adalah  $40 \text{ Hz} - 38.46 \text{ Hz} = 1.54 \text{ Hz}$ .



**Gambar 4. 5** Rangkaian Pengujian *Low Pass Filter*



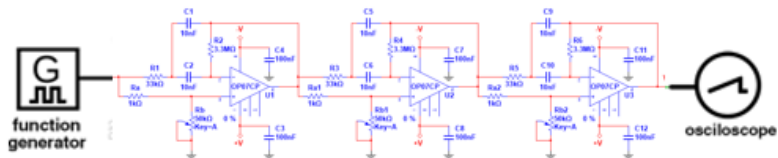
**Gambar 4. 6** Grafik Pengujian Rangkaian *Low Pass Filter*

**Tabel 4. 2** Pengujian *Low Pass Filter*

Finput(Hz)	Voutput(Vpp)
10	1
15	1
20	0.96
25	0.92
30	0.88
35	0.78
38.46	0.7
40	0.676
45	0.592
50	0.512
55	0.444

#### 4.1.4. Pengujian *Band Stop Filter*

Seperti filter sebelumnya, pengujian *Band Stop Filter* dilakukan dengan memberi sinyal masukkan sinyal sinus dengan amplitudo yang sama pada frekuensi berbeda beda. Frekuensi tegangan ini terletak di sekitar frekuensi *cutoff* sehingga dapat diketahui respon frekuensi dari rangkaian seperti pada Gambar 4.8. Rangkaian pengujian dapat dilihat pada Gambar 4.7 dan hasil pengujiannya pada Tabel 4. 3



**Gambar 4. 7** Rangkaian Pengujian *Band Stop Filter*

**Tabel 4. 3** Pengujian Rangkaian *Band Stop Filter*

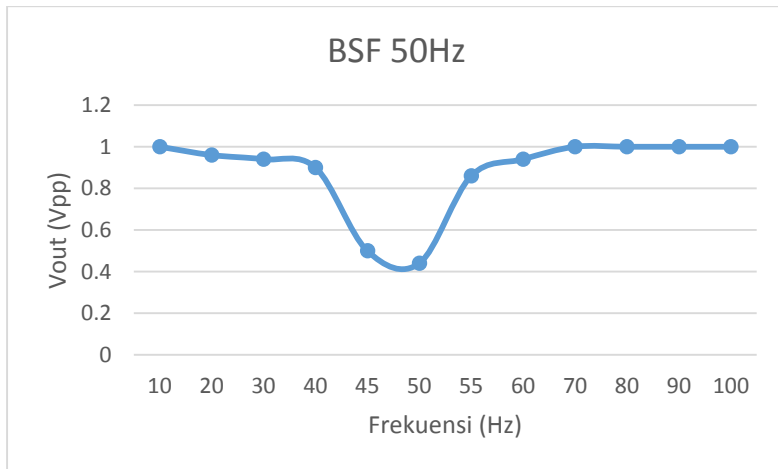
frekuensi input (Hz)	Vpp input (Volt)	Vpp output (Volt)
10	1	1
20	1	0,96
30	1	0,94
40	1	0,9
45	1	0,5
50	1	0,44
55	1	0,86
60	1	0,94
70	1	1
80	1	1
90	1	1
100	1	1

Berdasarkan pengujian dilihat bahwa *filter* dirancang mempunyai frekuensi resonansi 50 Hz dengan *bandwidth* 10Hz. Secara teori pada saat frekuensi input sama dengan frekuensi resonansinya nya diharapkan penguatannya paling minimum. Berdasarkan pengujian didapat

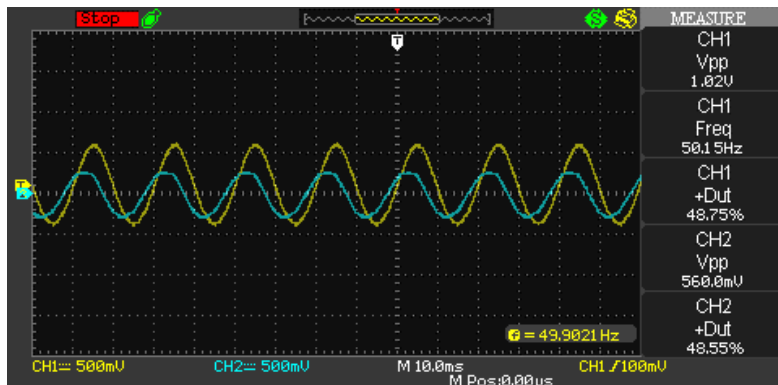
- Saat frekuensi input 50 Hz didapat penguatan 0,56 kali.
- *bandwidth* sebesar 10Hz.
- Penguatan maksimal pada daerah *passband* adalah 1 kali.
- 

#### **4.1.5. Pengujian Rangkaian Penjumlah**

Pengujian rangkaian penjumlah dilakukan dengan memberikan masukan berbagai macam sinyal dengan berbagai macam tegangan dan frekuensi dan dilihat kenaikan tegangannya. Tegangan *offset* tergantung pada tegangan baterai. Gambar rangkaian dapat dilihat pada Gambar 4.10.



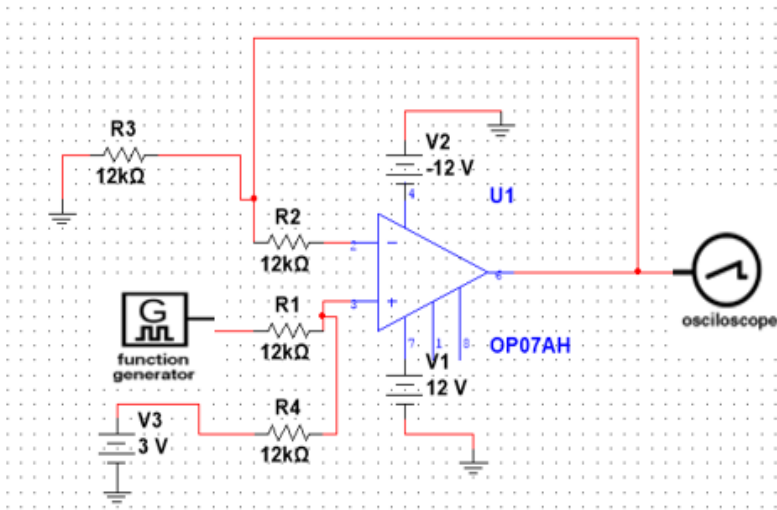
**Gambar 4. 8** Grafik Pengujian Rangkaian Band Stop Filter



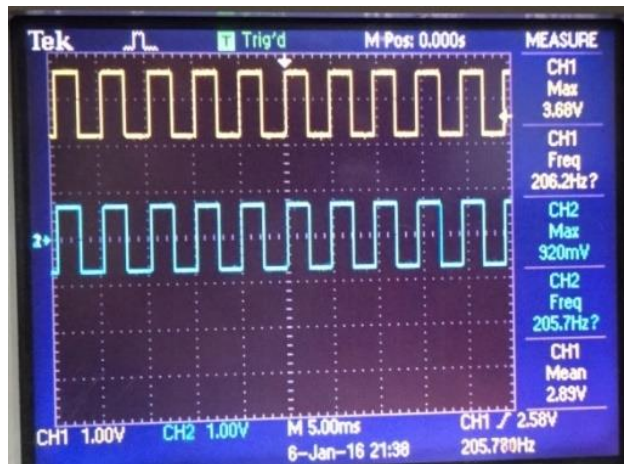
**Gambar 4. 9** Pengujian BSF ketika frekuensi 50Hz

Pada saat diberikan gelombang kotak, sinyal keluarannya memiliki bentuk yang sama dengan input, sehingga dapat dianalisa bahwa tegangannya tidak merubah frekuensi atau bentuk sinyal. Hasil pengujian dapat dilihat pada Gambar 4.11. Nilai penjumlahan dapat dikalkulasi sebagai menggunakan persamaan di bawah ini

$$V_{pp\ input} = 1,84\ V$$



**Gambar 4. 10** Rangkaian Pengujian Adder



**Gambar 4. 11** Hasil Pengujian Rangkaian Penjumlah



$$V_{pp\ output} = 1,86\ V$$

$$A_v = \frac{V_{out}}{V_{in}} = \frac{1,86}{1,84} = 1.01\ kali$$

$$V_{input} = 920\ mV$$

$$V_{output} = 3,68\ V$$

$$V_{adder} = 3,68 - 0,92 = 2,76\ V$$

## 4.2. Pengujian Sistem Piranti Lunak

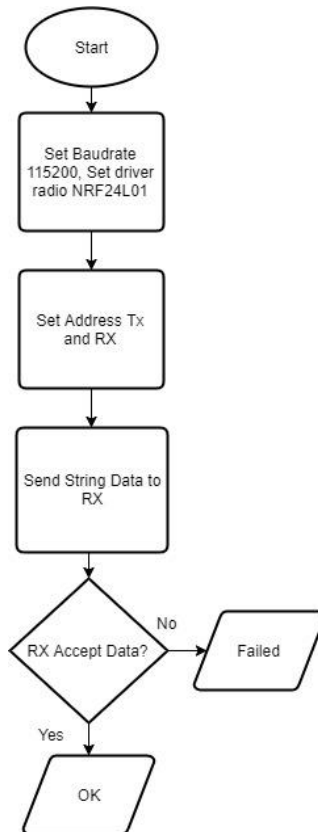
Setelah sistem perangkat keras elektrokardiogram, sistem piranti lunak juga diuji. Pengujian dilakukan dengan menguji jarak jangkauan NRF24L01, pengujian multinode(2 transmitter dan 1 receiver), pengujian keseluruhan sistem menggunakan GUI di Raspberry Pi.

### 4.2.1. Pengujian Jarak Cakupan NRF24L01

Pengujian menggunakan sepasang NRF24L01. 1 NRF24L01 sebagai transmitter dan 1 lainnya sebagai receiver. Cara pengujiannya dengan mengirimkan suatu string “Hello ini dari node 1” ke receiver. Apabila receiver menerima string tersebut, transmitter akan menerima ack “Ok”. Tetapi sebaliknya jika receiver tidak menerima data, transmitter akan menampilkan “failed”. Pengujian dilakukan dua kali di ruang terbuka tanpa halangan dari jarak 2 meter sampai 36 meter. Berikut flowchart program transmitter dan hasil pengujian pada jarak tertentu. Tabel pengujian jarak cakupan dapat dilihat pada Tabel 4.4.

### 4.2.2. Pengujian Multinode (2 Transmitter dan 1 Receiver)

Untuk penggunaan Multitransmitter pada NRF24L01 dibutuhkan library <RF24Network.h> dan peng-address-an transmitter dan receiver. Pengujian dilakukan dengan transmitter 1 mengirimkan string “hello ini dari node 1” dan transmitter 2 mengirimkan string “hello ini dari node 2”. Receiver akan menampilkan data string yang diterima. Pengujian pada serial monitor dapat dilihat pada Gambar 4.14.



**Gambar 4. 12** Flowchart pengujian jarak cakupan NRF24L01

#### **4.2.3. Pengujian Komputasi Data**

Tahapan selanjutnya adalah pengujian komputasi data. Pengujian dilakukan di program Delphi, karena di pengujian banyak grafik di Raspberry Pi membutuhkan program yang banyak. Berikut adalah hasil algoritma QRS detektor.

**Tabel 4. 4** Pengujian jarak cakupan NRF24L01

Jarak (m)	Jumlah fail/ jumlah pengiriman (pengujian 1)	Loss data 1	Jumlah fail/ jumlah pengiriman (pengujian 2)	Loss data 2
2	0/351	0%	0/458	0%
4	0/412	0%	0/413	0%
6	0/317	0%	0/354	0%
12	0/353	0%	0/348	0%
18	0/346	0%	0/355	0%
24	1/233	0.4%	5/293	1.7%
30	0/356	0%	0/353	0%
36	0/336	0%	6/380	1.5%

Pada tahapan pertama pada, sinyal disampling dan ditampilkan secara digital. Sinyal ini kemudian dilewatkan band pass filter 10 Hz sampai 20 Hz untuk mengisolasi frekuensi QRS kompleksnya. Dari hasil simulasi dapat dilihat bahwa algoritma sudah tepat karena proses tersebut menguatkan sinyal QRS dan melemahkan sinyal dan noise lainnya.

Pemrosesan yang kedua adalah dengan menggunakan algoritma derivatif atau penurunan. Dari hasil pengukuran dapat dilihat bahwa keluaran sinyal sama dengan teorema, yaitu merepresentasikan gradien dari sinyal. Namun dapat dilihat pula bahwa sinyal ini tidak banyak mempengaruhi morfologi sinyal karena keluarannya hampir serupa dengan filter band pass, sehingga tahapan ini juga dapat dihilangkan.

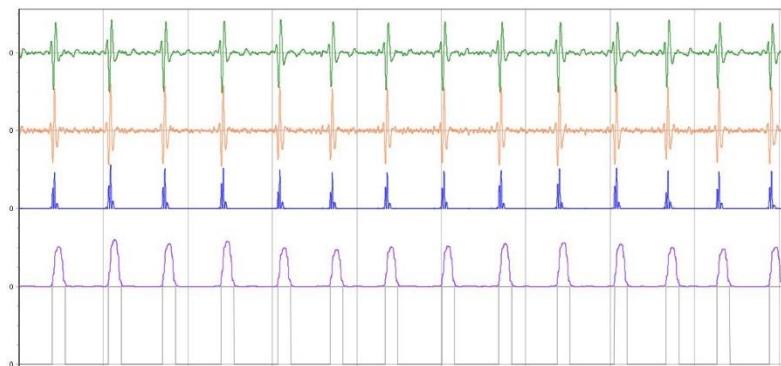
Pemrosesan selanjutnya adalah dengan fungsi kuadrat atau *squaring*. Fungsi ini membuat keluaran sinyal menjadi positif dan sekaligus memperbesar sinyal dengan amplitudo diatas 1, dan meredam



**Gambar 4. 13** Serial monitor receiver multinode pada jarak 12 meter

sinyal dengan amplitudo di bawah satu. Dapat dilihat bahwa teorema tersebut sesuai dengan hasil perhitungan.

Setelah dikuadratkan, proses yang terakhir sebelum dipotong adalah *moving average* atau *moving window integral*. Berdasarkan teorema ini sinyal diperhalus dengan penjumlahan rata rata. Pada



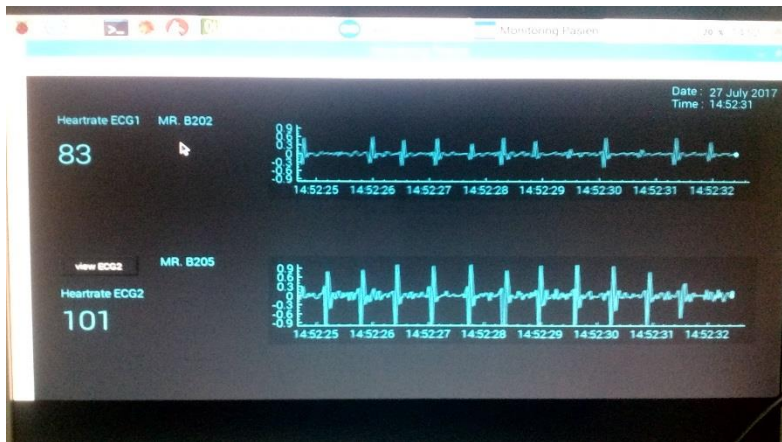
**Gambar 4. 14** Pengujian Algoritma Pan Tomkin di Delphi

algoritma ini digunakan *window* sebanyak 20, dan diperoleh hasil yang sesuai dengan teori.

Pengujian terakhir adalah dengan menguji algoritma *thresholding*. Pada bagian ini dideteksi tepi naik dan tepi turun sinyal hasil *moving window integral* dan dijadikan parameter logika tinggi atau rendah. Logika tinggi menyatakan segmen QRS. Setelah itu untuk setiap tepi naik *threshold* logika dibuat tinggi sebagai pertanda mulainya *R-R interval*. Pada hasil pengujian diperoleh hasil yang sama dengan teori. Berdasarkan analisa algoritma diatas dapat disimpulkan bahwa sinyal keluaran rangkaian sesuai dengan fungsi tiap proses sinyal.

#### 4.2.4. Pengujian Sistem Keseluruhan

Pengujian dilakukan dengan menggunakan 2 orang subjek untuk memakai elektrokardiogram, kemudian data yang didapat dari kedua elektrokardiogram dikirim melalui masing masing NRF24L01 ke receiver NRF24L01. Data dari receiver dikirim secara serial ke Raspberry Pi. Raspberry Pi sebagai pengolah *interface* data masing masing EKG. Setelah diplot, data diolah menggunakan algoritma pan tompkin dan RR interval untuk dihitung masing masing ekg. Pengujian tampilan di Raspberry Pi dapat dilihat pada Gambar 4.15.



**Gambar 4. 15** GUI Multinode EKG Monitoring di Raspberry Pi

*Halaman ini sengaja dikosongkan*

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan perancangan, pengujian, dan analisa sistem dapat ditarik beberapa kesimpulan dari tugas akhir ini. Pertama, penguatan instrumentasi menggunakan AD620 dengan diatur penguatan yang besar lebih dari 500 kali untuk memperbesar sinyal jantung yang memiliki orde milivolt. Rangkaian filter dengan orde 2 dan topologi *Sallen-Key* digunakan untuk meredam *noise*, oleh karena itu digunakan *high pass filter* dengan frekuensi *cutoff* 0,05Hz, *low pass filter* dengan frekuensi *cutoff* 38.46Hz, dan *band stop filter* dengan frekuensi 50Hz teredam sebesar 44%. Agar dapat diproses secara digital sinyal dinaikkan ke polaritas positif dengan memberi tegangan *offset* melalui *adder*.

Ketika sinyal jantung memasuki ADC Arduino, sinyal dikonversi menjadi digital dengan resolusi 10 bit. Data yang didapat dikirim melalui masing masing transmitter NRF24L01 ke receiver NRF24L01. NRF24L01 dapat digunakan maksimal sampai jarak 36 meter tanpa halangan dengan loss data 1.5%. Sinyal dari multinode elektrokardiogram tersebut diolah menggunakan *software Qt Creator* di Raspberry Pi.

Data yang didapat di QT Creator kemudian di pisah untuk masing masing ekg. Setelah dipisah, dilakukan perhitungan heart rate menggunakan algoritma pan tompkin dan RR interval.

#### **5.2. Saran**

Berdasarkan pengujian dapat diambil beberapa saran. Untuk pengembangan sistem kedepannya, aplikasi diharapkan dapat disempurnakan sehingga mampu mengeksekusi fungsi yang lebih kompleks seperti penambahan LCD grafik pada setiap node atau integrasi dengan sistem notifikasi dokter yang bersangkutan.

Dari sisi alat elektrokardiogram, dimensi alat diharapkan dapat diperkecil lagi dengan membuat rangkaian analog menjadi rangkaian SMD, dan dapat dikaji pula pemakaian filter digital untuk menggantikan filter analog.

Kekurangan menggunakan NRF24L01 adalah batas maksimal penggunaan node berdasar datasheetnya adalah 6 node, dan jarak cakupannya kurang memadai untuk digunakan saat beda ruangan.



Sehingga dapat dikaji penggunaan ethernet untuk memperbanyak node dan menambah jarak cakupan tiap ruangan.

## DAFTAR PUSTAKA

- [1] Badan Nasional Penanggulangan Bencana, “Data dan Informasi Bencana Indonesia”, BNPB, 2017
- [2] Tompkins, Willis, “*Biomedical Digital Processing*”, Prentice Hall, 2000
- [3] Tompkins, Willis, “*Biomedical Digital Signal Processing*”, Prentice Hall, 1995
- [4] Saryono, “*Cara Praktis Baca Elektrokardiografi*”, Nuha Medika, 2014
- [5] Guyton, Arthur C., “*Textbook of Medical Physiology*”, 11<sup>th</sup> edition Elsevier Inc., 2006
- [6] Pakpahan, H.A.P., “*Elektrokardiografi Ilustratif: Belajar EKG dengan Ilustrasi Sederhana*”, Badan Penerbit Fakultas Kedokteran Universitas Indonesia, 2012.
- [7] Martini., Nath., “*Fundamentals of Anatomy and Physiology*”, Ninth Edition, 2012
- [8] Pujiono, “*Rangkaian Elektronika Analog*”, Graha Ilmu, 2006
- [9] Setiawan, Rachmad, “*Teknik Akuisisi Data*”, Graha Ilmu, 2008
- [10] Hanif Messa “*Rancang Bangun Sistem Instrumentasi dan Pengolahan Digital Sinyal ECG untuk Analisa Variabilitas Parameter Temporal Berbasis Mikrokontroler ARM*”, Institut Teknologi Sepuluh Nopember, 2015
- [11] Setya, Vinda, “*Sistem Monitoring Heart Rate untuk Latihan Treadmill berbasis Mikrokontroler ARM dengan Antarmuka Nirkabel*”, Institut Teknologi Sepuluh Nopember, 2015
- [12] Fitriyatul, Nada, “*Analisis Multimodal Sinyal Jantung (ECG, PCG, dan Carotid Pulse) untuk Klasifikasi Jantung Normal dan Abnormal*”, Institut Teknologi Sepuluh Nopember, 2016
- [13] Wiranadi, Theo, “*Sistem Pengawasan Irama Jantung dan Indikasi Serangan Jantung dengan Elektrokardiografi Portabel yang Terintegrasi Android*”, Institut Teknologi Sepuluh Nopember, 2016
- [14] Dr. Ganesh V Bhat., Anandraddi Naduvanamani., “*Real Time ECG Acquisition System Using Raspberry Pi*”, IJESRT, June 2014
- [15] Özkan Kafadar, Adnan Sondas., “*Real Time Monitoring of ECG Signals using Raspberry Pi*”, IEEE, 2016

- [16] S.Ahmed, Hanadi., Ali, A. Abduladhem., *"Wireless Sensor Network for Medical Applications"*, Iraq J. Electrical and Electronic Engineering, Vol.11 No1, 2015
- [17] AD620 datasheet, Analog Device, 2010.
- [18] NRF24L01+ datasheet, Nordic Semiconductor, March 2008.

## LAMPIRAN

### Program Transmitter 1

```
#include <RF24Network.h>
#include <RF24.h>
#include <SPI.h>

RF24 radio(8,10);
RF24Network network(radio);
const uint16_t this_node = 01;
const uint16_t other_node = 00;
int data;

void setup(void)
{
  Serial.begin(115200);
  SPI.begin();
  radio.begin();
  network.begin(/*channel*/ 90, /*node address*/ this_node);
}

void loop() {

  network.update();
  RF24NetworkHeader header(other_node);
  data = analogRead(A0);
  bool ok = network.write(header,&data,sizeof(data));
  Serial.println(data);
  delay(20);
}
```

### Program Transmitter 2

```
#include <RF24Network.h>
#include <RF24.h>
#include <SPI.h>

RF24 radio(8,10);
RF24Network network(radio);
const uint16_t this_node = 02;
```

```

const uint16_t other_node = 00;
int data;

void setup(void)
{
  Serial.begin(115200);
  SPI.begin();
  radio.begin();
  network.begin(/*channel*/ 90, /*node address*/ this_node);
}

void loop() {

  network.update();
  RF24NetworkHeader header(other_node);

  data = analogRead(A0);
  bool ok = network.write(header,&data,sizeof(data));
  Serial.println(data);
  delay(20);
}

```

### **Program Receiver**

```

#include <RF24Network.h>
#include <RF24.h>
#include <SPI.h>

RF24 radio(8,10);
RF24Network network(radio);
const uint16_t this_node = 00;
const uint16_t other_node = 01;
static int pesan;
static int data[2];

void setup(void)
{
  Serial.begin(115200);
  SPI.begin();
  radio.begin();

```

```

network.begin(/*channel*/ 90, /*node address*/ this_node);
}

void loop(void){
  network.update();
  while ( network.available() ) {
    RF24NetworkHeader header;
    network.read(header,&pesan,sizeof(pesan));
    data[header.from_node-1] = pesan;
    if (header.from_node == 2) {

      Serial.print(data[0]);

      Serial.print(",");
      Serial.println(data[1]);
    }
  }
}

```

### **Program GUI di Raspberry Pi menggunakan Qt Creator**

```

#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <stdio.h>
#include <iostream>
#include <stdlib.h>
#include <string>
#include <unistd.h>
#include <errno.h>
#include <QLabel>
#include <QTextCodec>
#include <QDebug>
#include <QDesktopWidget>
#include <QScreen>
#include <QMessageBox>
#include <QMetaEnum>
#include <QtCore>
#include <QFile>
#include <QTextStream>
#include <QColor>

```

```

#include <QDateTime>
#include <QtMath>

char buff;
int ts = 0.02;
int ts2 = qPow(ts, 2);
int i =0, j =0, k=0, p =0,q=0, iold1=0, iold2 =0;
bool ok=false;
double v_ecg = 0.0, temp = 0.0;
double v_ecgdac = 0.0, tempdac = 0.0;
double tanda1 = 0.0, tanda2 = 0.0;
double sig1[100000], lpf1[100000], hpf1[100000], derf1[100000],
squ1[100000], win1[100000], thr1[100000];
double sig2[100000], lpf2[100000], hpf2[100000], derf2[100000],
squ2[100000], win2[100000], thr2[100000];
double T1[5], T2[5];
double T1avg, T2avg;
int heartrate1 = 0.0, heartrate2 = 0.0;

QSerialPort serial;
QFile *f, *f_sync;
QTextStream *inS;
QString file_name;
QStringList sig;
QByteArray datatest;

void initializeSerial(){
    serial.setPortName("ttyACM0");
    if (serial.open(QIODevice::ReadWrite)){
        if(!serial.setBaudRate(QSerialPort::Baud115200))
            qDebug()<<"baud error";
        if(!serial.setDataBits(QSerialPort::Data8))
            qDebug()<<"bits error";
        if(!serial.setParity(QSerialPort::NoParity))
            qDebug()<<"parity error";
    }else{
        qDebug() << "Serial cannt open";
    }
}

```

```

MainWindow::MainWindow(QWidget *parent)
: QMainWindow(parent)
, ui(new Ui::MainWindow)
{
    ui->setUpUi(this);
    //setWindowFlags(Qt::Window | Qt::WindowMinimizeButtonHint |
Qt::WindowCloseButtonHint);
    timer_time = new QTimer(this);
    timer = new QTimer(this);
    timer2 = new QTimer(this);
    TimerECG = new QTimer(this);
    TimerECGKwad = new QTimer(this);
    connect(timer_time, SIGNAL(timeout()), this, SLOT(set_time()));
    timer_time->start(1000);
    QCustomPlot *PlotECG = ui->customPlot;
    QCustomPlot *PlotKedua = ui->customPlot_2;
    setUpRealtimeDataDemo(PlotECG);
    setUpRealtimeDataDemo(PlotKedua);
}
MainWindow::~~MainWindow()
{
    delete ui;
}
void MainWindow::set_time(){
    QDateTime now = QDateTime::currentDateTime();
    QString s_date = now.toString("dd MMMM yyyy");
    QString s_time = now.toString("HH:mm:ss");
    ui->label_date->setText(s_date);
    ui->label_time->setText(s_time);
}
//klik mulai
void MainWindow::on_actionStart_Monitoring_triggered()
{
    initializeSerial();
    serial.clear();
}

```



```

connect(timer2, SIGNAL(timeout()), this, SLOT(getval()));
timer2->start(20);
connect(timer, SIGNAL(timeout()), this, SLOT(mulai()));
timer->start(1000);
connect(timer2, SIGNAL(timeout()), this, SLOT(runPlotECG()));
timer2->start(20);
connect(timer2, SIGNAL(timeout()), this,
SLOT(runPlotECGKuad()));
    timer2->start(20);
}
//stop
void MainWindow::on_actionStop_Monitoring_triggered()
{
    TimerECG->stop();
    TimerECGKuad->stop();
    TimerBreath->stop();
    timer->stop();
    timer2->stop();
    sig.empty();
    f->close();
    serial.close();
}

//data masuk
void MainWindow::getval(){
    if(serial.waitForReadyRead(30)){
        QString str = serial.readLine();
        i++;
        QStringList data_sensor = str.split(",");
        if(data_sensor.length()>=2){
            QDateTime rec = QDateTime::currentDateTime();
            QString time_t = rec.toString("yyyy-MM-dd HH:mm:ss");
            v_ecg = data_sensor.at(0).toDouble();
            temp = data_sensor.at(1).toDouble();
            v_ecgdac = (v_ecg*5.0/1024.0)-3.0;
            tempdac = (temp*5.0/1024.0)-1.8;

```

```

sig1[i] = v_ecgdac;
sig2[i] = tempdac;
//lpf20hz&hpf10hz
lpf1[i] = ((-11582.73*lpf1[i-1])-(8019.83*lpf1[i-
2])+(15791.36*sig1[i])+(31582.73*sig1[i-1])+(15791.36*sig1[i-
1]))/43562.89;
lpf2[i] = ((-11582.73*lpf2[i-1])-(8019.83*lpf2[i-
2])+(15791.36*sig2[i])+(31582.73*sig2[i-1])+(15791.36*sig2[i-
1]))/43562.89;

hpf1[i] = ((12104.31*hpf1[i-1])-(5062.07*hpf1[i-
2])+(10000*lpf1[i])-(20000*lpf1[i-1])+(10000*lpf1[i-2]))/22833.6;
hpf2[i] = ((12104.31*hpf2[i-1])-(5062.07*hpf2[i-
2])+(10000*lpf2[i])-(20000*lpf2[i-1])+(10000*lpf2[i-2]))/22833.6;

derf1[i] = (2*hpf1[i]+hpf1[i-1]-hpf1[i-3]-2*hpf1[i-4])/8;
derf2[i] = (2*hpf2[i]+hpf2[i-1]-hpf2[i-3]-2*hpf2[i-4])/8;
squ1[i] = qPow(derf1[i], 2)*100;
squ2[i] = qPow(derf2[i], 2)*100;
for(j = 1; j < 11; j++){
    win1[i] = win1[i]+((squ1[i-(10-j)])/10);
}
for(k = 1; k < 11; k++){
    win2[i] = win2[i]+((squ2[i-(10-k)])/10);
}
if (win1[i]>0.2){
    thr1[i]=1;
}else{
    thr1[i]=0;
}
if (win2[i]>0.6){
    thr2[i]=1;
}else{
    thr2[i]=0;
}

```

```

//bpm1
tanda1=thr1[i]-thr1[i-1];
if ((tanda1>0)&&(thr1[i]>=0.7)&&(thr1[i-1]<0.7)){
    T1[p]=i-iold1;
    p++;
    if(p==5){
        p=0;
    }
    T1avg=(T1[0]+T1[1]+T1[2]+T1[3]+T1[4])/5;
    iold1 =i;
    heartrate1=60/(T1avg*0.02);
}
//bpm2
tanda2=thr2[i]-thr2[i-1];
if ((tanda2>0)&&(thr2[i]>=0.7)&&(thr2[i-1]<0.7)){
    T2[q]=i-iold2;
    q++;
    if(q==5){
        q=0;
    }
    T2avg=(T2[0]+T2[1]+T2[2]+T2[3]+T2[4])/5;
    iold2 =i;
    heartrate2=60/(T2avg*0.02);
}
QString sHeartrate1 = QString::number(heartrate1);
QString sHeartrate2 = QString::number(heartrate2);
ui->h_rate->setText(sHeartrate1);
ui->h_rate_2->setText(sHeartrate2);
//ui->h_rate->setText(str);
}else{
    //qDebug() << "Time out";
}
}

//i++;
if (i==100000) {
    i=0;

```

```

    }
}

void MainWindow::on_pushButton_clicked()
{
    connect(TimerECGKuad, SIGNAL(timeout()), this,
    SLOT(runPlotECGKuad()));
    TimerECGKuad->start(20);
}

void MainWindow::setupRealtimeDataDemo(QCustomPlot
*customPlot)
{
    #if QT_VERSION < QT_VERSION_CHECK(4, 7, 0)
        QMessageBox::critical(this, "", "You're using Qt < 4.7, the realtime
data demo needs functions that are available with Qt 4.7 to work
properly");
    #endif
    demoName = "Real Time Data Demo";
    QPen colorTheme(QColor(38,198,218, 255));
    QColor colorTheme_(38,198,218, 255);
    customPlot->xAxis->setBasePen(colorTheme);
    customPlot->yAxis->setBasePen(colorTheme);
    customPlot->xAxis->setSubTickPen(colorTheme);
    customPlot->yAxis->setSubTickPen(colorTheme);
    customPlot->xAxis->setTickPen(colorTheme);
    customPlot->yAxis->setTickPen(colorTheme);
    customPlot->setBackground(QColor(32,32,32, 255));
    customPlot->xAxis->grid()->setVisible(false);
    customPlot->yAxis->grid()->setVisible(false);
    customPlot->addGraph(); // blue line
    customPlot->graph(0)->setPen(colorTheme);
    customPlot->addGraph(); // blue dot
    customPlot->graph(1)->setPen(colorTheme);
    customPlot->graph(1)->setLineStyle(QCPGraph::lsNone);
    customPlot->graph(1)->setScatterStyle(QCPScatterStyle::ssDisc);

```

```

customPlot->xAxis->setTickLabelType(QCPAxis::ltDateTime);
customPlot->xAxis->setDateTimeFormat("hh:mm:ss");
customPlot->xAxis->setAutoTickStep(false);
customPlot->xAxis->setTickStep(1);
customPlot->xAxis->setTickLabelColor(colorTheme_);
customPlot->yAxis->setTickLabelColor(colorTheme_);
// make left and bottom axes transfer their ranges to right and top axes:
connect(customPlot->xAxis, SIGNAL(rangeChanged(QCPRange)),
customPlot->xAxis2, SLOT(setRange(QCPRange)));
connect(customPlot->yAxis, SIGNAL(rangeChanged(QCPRange)),
customPlot->yAxis2, SLOT(setRange(QCPRange)));
}

//-----Run Plot ECG Original-----//
void MainWindow::runPlotECG()
{
    // calculate two new data points:
#ifdef QT_VERSION < QT_VERSION_CHECK(4, 7, 0)
    double key = 0;
#else
    double key =
QDateTime::currentDateTime().toMSecsSinceEpoch()/1000.0;
#endif
    QString adc1 = QString::number(v_ecg);
    // ui->h_rate->setText(adc1);
    double hrt = sig1[i];
    static double lastPointKey = 0;
    if (key-lastPointKey > 0.01) // at most add point every 10 ms
    {
        // add data to lines:
        ui->customPlot->graph(0)->addData(key, hrt);
        //ui->customPlot->graph(1)->addData(key, value1);
        // set data of dots:
        ui->customPlot->graph(1)->clearData();
        ui->customPlot->graph(1)->addData(key, hrt);
    }
}

```

```

// remove data of lines that's outside visible range:
ui->customPlot->graph(0)->removeDataBefore(key-8);
//ui->customPlot->graph(1)->removeDataBefore(key-8);
// rescale value (vertical) axis to fit the current data:
ui->customPlot->graph(0)->rescaleValueAxis();
//ui->customPlot->graph(1)->rescaleValueAxis(true);
lastPointKey = key;
}
// make key axis range scroll with the data (at a constant range size of
8):
ui->customPlot->xAxis->setRange(key+0.25, 8, Qt::AlignRight);
ui->customPlot->yAxis->setRange(-1,1);
ui->customPlot->replot();
// calculate frames per second:
static double lastFpsKey;
static int frameCount;
++frameCount;
if (key-lastFpsKey > 2) // average fps over 2 seconds
{
    lastFpsKey = key;
    frameCount = 0;
}
}
//-----Run Plot ECG filter-----//
void MainWindow::runPlotECGKuad()
{
    // calculate two new data points:
#ifdef QT_VERSION < QT_VERSION_CHECK(4, 7, 0)
    double key = 0;
#else
    double key =
QDateTime::currentDateTime().toMSecsSinceEpoch()/1000.0;
#endif
    double hrt = sig2[i];
    static double lastPointKey = 0;
    if (key-lastPointKey > 0.01) // at most add point every 10 ms

```

```

{
    // add data to lines:
    ui->customPlot_2->graph(0)->addData(key, hrt);
    //ui->customPlot->graph(1)->addData(key, value1);
    // set data of dots:
    ui->customPlot_2->graph(1)->clearData();
    ui->customPlot_2->graph(1)->addData(key, hrt);
    // remove data of lines that's outside visible range:
    ui->customPlot_2->graph(0)->removeDataBefore(key-8);
    //ui->customPlot->graph(1)->removeDataBefore(key-8);
    // rescale value (vertical) axis to fit the current data:
    ui->customPlot_2->graph(0)->rescaleValueAxis();
    //ui->customPlot->graph(1)->rescaleValueAxis(true);
    lastPointKey = key;
}
// make key axis range scroll with the data (at a constant range size of
8):
ui->customPlot_2->xAxis->setRange(key+0.25, 8, Qt::AlignRight);
ui->customPlot_2->yAxis->setRange(-1, 1);
ui->customPlot_2->replot();
// calculate frames per second:
static double lastFpsKey;
static int frameCount;
++frameCount;
if (key-lastFpsKey > 2) // average fps over 2 seconds
{
    lastFpsKey = key;
    frameCount = 0;
}
}

```

## BIODATA PENULIS



Penulis memiliki nama lengkap Dwi Sandi Wijayanto. Penulis yang disapa Sandi ini lahir di Surabaya. 02 Juli 1995 sebagai anak kedua dari pasangan Sukatmidjo dan Kartini. Penulis menempuh dan menamatkan pendidikan formal di SDN Dukuh Kupang V/534 Surabaya, SMPN 3 Surabaya, dan SMAN 5 Surabaya.

Pada tahun 2013 setelah lulus SMA, penulis menempuh pendidikan sarjana strata di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Selama perkuliahan penulis aktif dalam kegiatan BEM FTI dan berbagai kegiatan kepanitiaan.

Pada saat buku tugas akhir ini ditulis penulis telah mengambil konsentrasi bidang teknik biomedika pada bidang studi elektronika, dan sedang menyelesaikan tugas akhir pada semester 8 perkuliahan.

Penulis dapat dihubungi melalui [dwisandidwisandi@gmail.com](mailto:dwisandidwisandi@gmail.com) atau [dwi13@mhs.ee.its.ac.id](mailto:dwi13@mhs.ee.its.ac.id)



